ForestQC: quality control on genetic variants from next-generation sequencing data using random forest

Jiajin Li[1], Brandon Jew[2], Lingyu Zhan[3], Sungoo Hwang[4], Giovanni Coppola[4], Nelson B. Freimer[1,4], Jae Hoon Sul[4,*]

1. Department of Human Genetics, David Geffen School of Medicine, University of California, Los Angeles, CA 90095, USA
2. Interdepartmental Program in Bioinformatics, University of California, Los Angeles, CA 90095, USA
3. Molecular Biology Institute, University of California, Los Angeles, CA 90095, USA
4. Department of Psychiatry and Biobehavioral Sciences, University of California, Los Angeles, CA 90095, USA
*. Email: jaehoonsul@mednet.ucla.edu

ABSTRACT

Next-generation sequencing (NGS) technology enables discovery of nearly all genetic variants present in a genome. A subset of these variants, however, may have poor sequencing quality due to limitations in sequencing technology or in variant calling algorithms. In genetic studies that analyze a large number of sequenced individuals, it is critical to detect and remove those variants with poor quality as they may cause spurious findings. In this paper, we present a statistical approach for performing quality control on variants identified from NGS data by combining a traditional filtering approach and a machine learning approach. Our method uses information on sequencing quality such as sequencing depth, genotyping quality, and GC contents to predict whether a certain variant is likely to contain errors. We applied our method to two whole-genome sequencing datasets where one dataset consists of related individuals from families while the other consists of unrelated individuals. Results indicate that our method outperforms widely used methods for performing QC on variants such as VQSR of GATK by considerably improving the quality of variants to be included in the analysis. Our approach is also very efficient, and hence can be applied to large sequencing datasets. We conclude that combining a machine learning algorithm trained with sequencing quality information and the filtering approach is an effective approach to perform quality control on genetic variants from sequencing data.

Introduction

Over the past few years, genome-wide association studies (GWAS) have been playing an important role in identifying genetic variations associated with common diseases or complex traits [1, 2]. GWAS have found many associations between common variants and human diseases, such as schizophrenia[3], type 2 diabetes[4, 5] and Parkinson's Disease[6]. However, these common variants typically explain only a small fraction of heritability for the complex traits[7, 8]. Rare variants are another type of genetic variants that have been considered as an important risk factor for complex traits and common diseases[9, 10, 11, 12]. With the next generation sequencing (NGS) technology, geneticists may now gain insights into the roles of novel or rare variants. For instance, deep targeted sequencing was applied to discover rare variants associated with inflammatory bowel disease[13]. Whole genome sequencing (WGS) has been used to identify rare variants associated with prostate cancer[14], and with whole exome sequencing, studies have also detected rare variants associated with LDL cholesterol[15] and autism[16].

NGS data are not, however, perfect, and the quality of variants detected by sequencing may be adversely influenced by several factors. First, genome sequencing is known to have errors or biases[17, 18, 19, 20, 21], which might cause inaccuracy in detecting variants. Second, sequence mappability of different regions may not be uniform, but correlated with sequence-specific biological features, leading to alignment biases. For instance, it is shown that introns have significantly lower mappability levels than exons[22]. Third, variant calling algorithms may be sources of errors as no algorithm is 100% accurate. For example, GATK HaplotypeCaller and GATKUnifiedGenotyper[23], which are the widely used variant callers, have sensitivity of about 96% and precision of about 98%[24]. Additionally, different variant callers may generate discordant calls on some variants[25], which indicates inaccuracy of those calls, and in certain cases, different versions of even the same software may generate inconsistent calls. All these factors may generate false positive sites or incorrect genotypes, which may then lead to false positive associations in the follow-up association test. For example, Alzheimer's Disease Sequencing Project reports that they found spurious associations in the case-control analysis where one of the causes for the problem could be inconsistent variant calling processes for sequenced samples[26].

It is extremely important to perform quality control (QC) on genetic variants identified from sequencing to remove variants that may contain sequencing errors and hence are likely to be false positive calls. Traditionally, genetic studies have utilized two types of QC approaches; we call them, "filtering" and "classification" approaches. In filtering approaches, several filters are applied to remove problematic variants such as variants with high genotype missing rate (e.g. > 5%), low Hardy-Weinberg Equilibrium (HWE) p-value (e.g. < 1E-4), or very high or low allele balance of heterozygous calls (ABHet) (e.g. > 0.75 or < 0.25). One main problem with this type of approaches is that these thresholds are arbitrarily determined without strong statistical justification. We may also remove variants whose metrics are very close to the thresholds (e.g. variants with missing rate of 5.1%). Another type of QC is a classification approach that attempts to learn variants with low quality using machine learning approaches. One example is VQSR of GATK[24,27] that uses a Gaussian mixture model to learn the multidimensional annotation profile of variants with high and low quality. However, one of issues with VQSR is that one needs training datasets acquired from existing databases on variants such as 1000 Genomes Project[29] and HapMap[30], which may be biased to keep known variants and filter out novel variants. Another issue is that those known databases of genetic variants may not be always accurate, which would lead to inaccurate classification of variants, and they may not even be available for some species. It may also be a challenge to apply VQSR to a variant call set generated by variant callers other than GATK as VQSR needs metrics of variants that are not often calculated by non-GATK variant callers.

In this article, we present ForestQC for performing QC on genetic variants discovered through sequencing. Our method aims to identify whether a specific variant is of high sequencing quality (“good” variants) or of low quality (“bad” variants) by combining the filtering and classification approaches. We first apply a filtering approach to detect obviously good and bad variants from data. We use stringent filters such that those variants are truly good or bad while the rest of variants that are neither good nor bad are considered to have ambiguous quality (“gray” variants). Given this set of good and bad variants, we train a machine learning model whose goal is to classify whether gray variants are good or bad. With an insight that good variants would have higher genotype quality and sequencing depth than do bad variants, we use information of several sequencing quality measures of variants for model training. ForestQC then uses sequencing quality measures of gray variants to predict whether each gray variant has high or low sequencing quality. Our approach is different from the filtering strategy in that it only uses filters to identify truly good or bad variants and does not attempt to classify gray variants with filters. Our approach is also different from VQSR as our training strategy allows us to train our model without known datasets for variants and solves several aforementioned issues with VQSR. Another advantage of our approach is that it can be applied to standard Variant Call Format (VCF) files from any variant callers and is very efficient.

To demonstrate accuracy of ForestQC, we apply it to two high-coverage WGS datasets; 1) large extended pedigrees ascertained for bipolar disorder (BP) from Costa Rica and Colombia[31], and 2) a sequencing study for Progressive Supranuclear Palsy (PSP). The first dataset includes 449 related individuals from families while the latter dataset consists of 495 unrelated individuals. We show that ForestQC outperforms VQSR and a filtering approach based on ABHet as good variants detected from ForestQC have higher sequencing quality than those from VQSR and the filtering approach in both datasets. This suggests that our approach identifies high-quality variants more accurately than other approaches in both family and unrelated datasets. ForestQC is publicly available at https://github.com/avallonking/ForestQC

Results

Overview of ForestQC

ForestQC takes a raw VCF file as input and determines whether each variant has “good” sequencing quality or “bad” quality. Our method combines a filtering approach that determines good and bad variants by a set of pre-defined filters and a classification approach that uses machine learning to classify whether a variant is good or bad. As illustrated in Figure 1, our method first calculates statistics of each variant for several filters that are commonly used in performing QC in GWAS. These statistics consist of ABHet, HWE p-value, genotype missing rate, Mendelian error rate for family data, and any user-defined statistics (details described in Method session). ForestQC then identifies three sets of variants using these statistics for filters: 1) a set of good variants that pass all filters, 2) a set of bad variants that clearly fail filter(s), and 3) a set of gray variants that are neither good nor bad variants. We use stringent thresholds for filters (Supplementary Table 1 and 2), and hence we are highly confident that good variants are of high quality while bad variants are truly false positives or have unequivocally poor sequencing quality. The next step in ForestQC is to train a random forest machine learning model using the good and bad variants we detect from the filtering step. In ForestQC, seven sequencing quality metrics of good and bad variants are used as features to train the random forest model, including three related to sequencing depth, three related to genotype quality, and one related to the GC content. Finally, the fitted model predicts whether each gray variant is good or bad. We combine the predicted good variants from the random forest model and the good variants from the filtering step, and they are all good variants determined by ForestQC. The same procedure is applied to identify bad variants.

One major challenge in classifying gray variants is to identify a set of sequencing quality metrics that are used as features to train the random forest model. We choose three sets of features based on quality metrics that variant callers provide and prior knowledge in genome sequencing. The first set of features is genotype quality (GQ) where we have three metrics: mean, standard deviation (SD), and outlier ratio. The outlier ratio is the proportion of samples whose GQ scores are lower than a particular threshold, and it measures a fraction of individuals who are poorly sequenced at a mutation site. A good variant is likely to have high mean, low SD, and low outlier ratio of GQ values. The second set of features is sequencing depth (DP) as low depth often introduces sequencing biases and reduces variant calling sensitivity[32]. We also use the same three sets of metrics for DP as those for GQ: mean, SD, and outlier ratio. The last set of features is related to genomic characteristics instead of sequencing quality, which is GC content. High or low GC content may decrease the coverage of certain regions[33, 34] and thus may lower the quality of variant calling. Hence, a good variant would have moderate GC content. Given these three sets of features, ForestQC learns how those features determine good and bad variants and classifies gray variants according to rules that it learns.

Comparison of different machine learning algorithms

As there are many different machine learning algorithms available, we first seek to find the most accurate and efficient algorithm for performing QC on NGS variant data. To ensure the quality of training and prediction, we choose supervised learning algorithms rather than unsupervised algorithms. Several major types of supervised algorithms are selected for comparison: random forest, logistics regression, k nearest neighbors (KNN), Naive Bayes, quadratic discriminant analysis (QDA), AdaBoost, artificial neural network (ANN), Gaussian mixture, and support vector machine (SVM). We use the BP WGS dataset, which consists of large pedigrees from Costa Rica and Colombia, to compare the performance of different algorithms. We use the aforementioned three sets of features related to sequencing quality for all algorithms we test. We apply the filtering approach (Supplementary Table 1 and 2) to the BP data to identify good, bad, and gray variants, and we choose 100,000 good and 100,000 bad variants randomly for model training. We then choose another 100,000 good and 100,000 bad variants randomly from the rest of variants for model testing. Each learning algorithm will be trained with the same training set and tested with the same test set. We use F1-score to measure classification accuracy during model testing, which is the harmonic average of precision (positive predictive value) and recall (sensitivity). The closer F1-score is to 1, the better the performance is. To assess efficiency of each algorithm, we measure its CPU time and real time during training and predicting. CPU time measures the total time across all threads and all CPUs while real time measures the clock time needed for executing an algorithm. For algorithms that cannot be parallelized, CPU time is supposed to be the same as real time. We use eight threads for algorithms that support parallelization.

Results show that random forest is the most precise model in SNV classification with F1-score of 0.97, and the second most accurate model in indel classification with F1-score of 0.94 (Table 1). Its CPU time is 89.64 seconds while its real time is 16.65 seconds in model training and prediction (Table 1), which ranks as the fifth fastest algorithm in terms of real time and the sixth fastest algorithm in terms of CPU time. As random forest randomly divides the entire dataset into several subsets of the same size and constructs decision trees independently in each subset, it is highly parallelizable, and it has low error rates and high robustness with respect to noise[35]. As for other learning algorithms, SVM and ANN are highly accurate (both with F1-score of 0.97 in SNV classification) although they are not as efficient as random forest. Especially, SVM is the slowest algorithm because of its inability to parallelize, which is about 100x slower than random forest in real time and 20x slower in CPU time (Table 1). This suggests that it may be computationally very expensive to use SVM in large-scale WGS datasets that have tens of millions of variants. Logistic regression, Naive Bayes and QDA are more efficient than random forest, but their predictions are not as accurate as those of random forest. For example, Naive Bayes needs only 0.23 seconds for training and prediction while its F1-score is the second lowest among all algorithms (0.90 and 0.87 in SNV and indel classification, respectively) (Table 1). This result demonstrates that random forest is both accurate and efficient, and hence we use it as the machine learning algorithm in our approach. To further improve the random forest algorithm, we test a different number of trees in the algorithm and we find that random forest with 50 trees balances efficiency and accuracy (Supplementary Figure 1). To identify good variants from gray variants, we use the probability of each gray variant being a good variant calculated from random forest, and we consider gray variants with the probability of being good variants > 50% as good variants as this probability threshold achieves the highest F1-score (Supplementary Figure 2).

Measuring performance of QC methods on WGS data

To evaluate the accuracy of ForestQC and other methods on WGS data, we calculate several statistics. For a family dataset, we calculate Mendelian error rate (ME) of each variant, which measures inconsistency in genotypes between parents and offspring. Another statistic we measure is genotype discordance rate between microarray and sequencing if individuals who are sequenced are also genotyped. In both WGS datasets we analyze, microarray data are available. These two statistics are important indicators of quality of variants because good variants would follow Mendelian inheritance patterns and their genotypes would be consistent between microarray and sequencing. In addition to these statistics, we measure several other statistics that are reported in sequencing studies such as the number of variants (SNVs and indels), transitions/transversions (Ti/Tv) ratio, the number of multi-allelic variants, genotype missing rate. We compute these QC-related statistics separately for SNVs and indels. We use these statistics to compare the performance of ForestQC with that of three approaches. The first is one without performing any QC (no QC). The second method is VQSR which is a classification approach that requires known truth sets for model training, such as HapMap or 1000 genomes. We use recommended resources and parameter settings to run VQSR as of 2018-04-04[36], but we also look at different settings. The third method is an ABHet approach, which is a filtering approach that retains variants according to allele balance of variants (see Methods).

Performance of ForestQC on family WGS data

We apply ForestQC to the BP WGS dataset that consists of 449 subjects with the average coverage of 36. There are 25.08 million (M) SNVs and 3.98M indels[31]. The variant calling is performed with GATK-HaplotypeCaller v3.5. This is an ideal dataset for assessing the performance of different QC methods because this dataset contains individuals from families who are both sequenced and genotyped. This study design allows us to calculate both ME rate and genotype discordance rate of variants between WGS and microarray. For this dataset, we test ForestQC with two different filter settings, one using ME rate as a filter and the other not using ME as a filter. The results of the former approach would filter out bad variants based on ME rate, and hence ME rate of good variants would be very low. However, we observe that both approaches have similar performance in terms of ME rate and other statistics (Supplementary Table 3, Supplementary Figures 3-4), and hence we show results of only ForestQC using ME rate as a filter.

Results show that ForestQC outperforms ABHet and VQSR in terms of the quality of good SNVs while it detects fewer good SNVs than the other approaches. ForestQC identifies 22.23M (88%) good SNVs, which is fewer than 22.42M (89%) and 24.11M (96%) good SNVs from ABHet and VQSR, respectively (Table 2). However, ABHet has 3.57x and VQSR has 9.64x higher ME rate on good SNVs than ForestQC (Figure 2 a), and ABHet has 1.50x and VQSR has 1.61x higher genotype discordance rate on good SNVs than ForestQC (Figure 2 b). In addition, ABHet and VQSR have 81.48x and 96.32x higher genotype missing rate on good SNVs than ForestQC, respectively (Figure 2 c), but it is important to note that genotype missing rate is used as a filter in ForestQC, which means SNVs with high genotype missing rate are filtered out. We observe that VQSR and ABHet have 313 thousand (K) (1.30%) and 235K (1.05%) good SNVs with very high genotype missing rate (>10%), respectively, and there are also 115K (0.48%, GATK) and 53K (0.24%, ABHet) good SNVs with very high ME rate (>15%) while ForestQC has none of them due to its filtering approach. The better quality of good SNVs from ForestQC means that bad SNVs detected from ForestQC would have lower quality, and results show that bad SNVs detected by our method have higher genotype missing rate and higher ME rates than those of ABHet and VQSR (Supplementary Figure 5 a, b). Interestingly, ForestQC does not have the highest genotype discordance rate on bad SNVs (Supplementary Figure 5 c). This may be due to the small number of bad SNVs genotyped with microarray (5,885, 2607, and 354 such SNVs for ForestQC, ABHet, and VQSR, respectively), and hence the reported average genotype discordance rate on bad SNVs may not be accurate. The no QC method keeps the greatest number of good SNVs (25.08M), but they have the highest ME rate, genotype missing rate, and genotype discordance rate as expected.

Next, we obtain several statistics of good SNVs commonly used in sequencing studies to evaluate the performance of ForestQC. One such statistic is Ti/Tv ratio, which is expected to be around 2.0 over the whole genome[37]. If this ratio is smaller than 2.0, it means that there may be false positive variants in the dataset. We compute Ti/Tv ratio for each individual across all good SNVs and look at the distribution of those ratios across all individuals (sample-level statistics). We find that the mean Ti/Tv ratio of good known SNVs (present in dbSNP) is around 2.0 for all four methods, which suggests that they have similar accuracy on known SNVs in terms of Ti/Tv ratio (Supplementary Figure 6 a). However, results show that the mean Ti/Tv ratio of good novel SNVs (not in dbSNP) from ForestQC is better than that of those SNVs from other methods; the mean Ti/Tv ratio is 1.68 for ForestQC, which is closest to 2.0 among other methods (1.42 for VQSR, 1.53 for ABHet, and 1.29 for No QC) (Figure 3 a). Paired t-tests for the difference in the mean Ti/Tv ratio between ForestQC and other methods are all significant (p-value < 2.2e-16 versus all other methods). This result suggests that novel SNVs predicted to be good by ForestQC are more likely to be true positives than those SNVs from other QC methods. Another statistic commonly used in sequencing studies is the percentage of multi-allelic SNVs, which are variants with more than one alternative allele. Given this sample size (449), many of them are likely to be false positives, and ForestQC has 32.69% and 42.62% smaller fraction of multi-allelic SNVs among good SNVs than do VQSR and no QC methods while the ABHet approach has the smallest fraction of such SNVs (Table 2).

In addition to SNVs, we apply the four QC methods to indels. Similar to results of SNVs, ForestQC identifies fewer good indels than does VQSR, but the quality of those indels from ForestQC is better than that of good indels from ABHet and VQSR. Out of total 3.98M indels, ForestQC predicts 2.79M indels (70%) to have good sequencing quality while VQSR and ABHet find 3.14M (79%) and 2.67M (67%) good indels, respectively (Table 2). Good indels from VQSR and ABHet, however, have 7x and 3x higher ME rate, and 17x and 29x higher genotype missing rate, than those from ForestQC, respectively (Figure 2 d, e). Bad indels identified by ForestQC have 1.68x and 1.32x higher ME rate, and 1.23x and 2.36x higher genotype missing rate than those from VQSR and ABHet, respectively (Supplementary Figure 5 d, e). Besides, we observe that there are 72K (2.29%, GATK) and 86K (3.23%, ABHet) good indels with very high genotype missing rate (>10%) and also 134K (4.26%, GATK) and 44K (1.66%, ABHet) good indels with very high ME rate (>15%) while there are no such indels in ForestQC. This result suggests that many good indels detected by ABHet or VQSR may be false positives or indels with poor sequencing quality. One of the reasons why VQSR does not perform well on indels could be the database it uses for training its machine learning model as VQSR considers all indels found in the database (Mills gold standard call set[38] and 1000G Project[39]) to be true variants. This leads VQSR to have a significantly higher proportion of known indels among good indels (85%), compared with 80% from ForestQC and 82% from ABHet (Table 2). The poor performance of VQSR on indels may be because not all indels in the database are true variants, or because even if they are true indels, those indels would not necessarily have high sequencing quality in the sequencing dataset of interest. Hence, this result demonstrates one of the limitations of using known databases for finding good variants. It is also important to note that in general, indels have much higher ME rate (0.41% for no QC) than that of SNVs (0.08% for no QC), which is expected given the greater difficulty of calling indels.

Another major difference between ForestQC and the other approaches is the allele frequency of variants after QC as ForestQC keeps a greater number of rare variants in its good variant set. Our method has 1.62% and 1.77% higher proportion of rare SNVs, and 5.30% and 13.16% higher proportion of rare indels than ABHet and VQSR do, respectively (Supplementary Table 4). We also observe this phenomenon in the variant-level and sample-level statistics for the number of SNVs. The variant-level statistics show that the number of good SNVs detected by ForestQC is similar to those from ABHet (Table 2). However, the sample-level statistics show that each individual on average carries fewer alternative alleles of good SNVs from ForestQC (3.58M total SNVs) than those from VQSR and ABHet (3.95M and 3.77M total SNVs, respectively) (Figure 3 b, c, Supplementary Figure 6 b). We observe a similar phenomenon for indels between ABHet and ForestQC (Table 2, Figure 3 d, Supplementary Figure 6 c, d). This phenomenon could be explained by the higher fraction of rare variants among good variants from ForestQC, as individuals would carry fewer variants if there are a greater fraction of rare variants. One main reason why ForestQC has the higher proportion of rare variants is that common variants have higher ME rate, genotype discordance rate and genotype missing rate than do rare variants (Supplementary Figure 7); because common variants are more heterozygous, it is more difficult to accurately call them. This suggests that while a majority of common variants may be true variants, some of them may not necessarily have high sequencing quality, and hence their calls may not be accurate enough for downstream analyses.

Performance of ForestQC on WGS data with unrelated individuals

To evaluate the performance of ForestQC on WGS datasets that contain only unrelated individuals, we apply it to the PSP dataset that has 495 individuals who are whole-genome sequenced at average coverage of 29, generating 33.27M SNVs and 5.09M indels. Among the 495 individuals who are sequenced, 381 individuals (77%) of them are also genotyped with microarray, which enables us to check the genotype discordance rate between WGS and microarray data. Because the PSP dataset contains only unrelated individuals, we do not report ME rate. Similar to BP WGS data, we apply four methods (ForestQC, VQSR, ABHet, and No QC) to the PSP dataset, although the parameter setting of VQSR has slightly changed. As the PSP dataset is called with GATK v3.2, the StrandOddsRatio (SOR) information from the VCF file is missing, which is recommended to use in VQSR, and hence this annotation is excluded from VQSR. However, we find that SOR information has little impact on the results of VQSR as we test VQSR without SOR information using the BP dataset and obtain similar results with one using SOR information (Supplementary Figure 8).

Similar to the results of the BP dataset, ForestQC identifies good variants with higher quality although it detects fewer good variants than other approaches. ForestQC identifies 29.25M (88%) good SNVs, which is slightly fewer than 29.77M (89%) good SNVs from ABHet but about 2 million fewer than 31.28M (94%) good SNVs from VQSR (Table 3). However, good SNVs from ABHet and VQSR have 53.76x and 42.55x higher genotype missing rate than those from ForestQC, respectively (Figure 4 a), but it is important to note that missing rate is included as a filter in ForestQC. In addition, there are 311K (0.99%, GATK) and 331K (1.13%, ABHet) good SNVs with very high genotype missing rate (>10%), while ForestQC removes all these SNVs. We also observe that bad SNVs from ForestQC have 2.4x higher genotype missing rate than those from ABHet, although bad SNVs from GATK have slightly higher missing rate than those from ForestQC (Supplementary Figure 9 a). Good SNVs from ABHet and VQSR have 1.28x and 1.29x higher genotype discordance rate than those from ForestQC, respectively (Figure 4 b). As for the genotype discordance rate of bad SNVs, both ABHet and VQSR have higher genotype discordance rate than does ForestQC (Supplementary Figure 9 b), but this may be due to the small number of bad SNVs genotyped with microarray (10,130, 4,121, and 553 such SNVs for ForestQC, ABHet, and VQSR, respectively). The variant-level and sample-level statistics also demonstrate the better quality of good SNVs from ForestQC. Although all methods have mean Ti/Tv ratio of good known SNVs above 2.0, the mean Ti/Tv ratio of good novel SNVs among all sequenced individuals is 1.65 for ForestQC, which is closer to 2.0 than other methods (1.27, 1.54, and 1.24 for VQSR, ABHet, no QC, respectively). (Supplementary Figure 10 a, Figure 5 a). Paired t-tests for the difference in the mean Ti/Tv ratio between ForestQC and other methods are all significant (p-value < 2.2e-16 versus all other methods). ForestQC has 16.67% and 33.33% smaller fraction of multi-allelic SNVs among good SNVs than do VQSR and no QC methods, respectively, while the ABHet approach has the smallest proportion of such SNVs (Table 3). Lastly, consistent with the results of the BP dataset, the sample-level statistics show that each individual on average carries fewer alternative alleles of good SNVs from ForestQC than those from VQSR and ABHet (Figure 5 b, c, Supplementary Figure 10 b). Rare SNVs in good SNVs from ForestQC account for 1.70% and 1.32% higher proportion, compared with those from ABHet and VQSR (Supplemental Table 5). This may be because rare SNVs have lower genotype missing rate and genotype discordance rate than do common variants (Supplementary Figure 11, a, b).

For indels, our method predicts 3.42M indels (67% of total 5.09M indels) to be good variants, which is slightly more than 3.31M (65%) good indels from ABHet and fewer than 3.68M (72%) good indels from VQSR (Table 3). Because the PSP dataset lacks ME rate as it contains only unrelated individuals and indels are not called in microarray, it is difficult to compare the performance of the QC methods on indels. We find that good indels from ABHet and VQSR have 27.02x and

18.77x higher genotype missing rate than those from our method, respectively (Figure 4 c). Additionally, VQSR and ABHet have 107K (2.91%) and 131K (4.08%) good indels with high genotype missing rate (>10%), respectively while ForestQC filters out all of these indels. Also, bad indels from ForestQC have 2.05x and 1.21x higher genotype missing rate than those from ABHet and VQSR, respectively (Supplementary Figure 9 c). This, however, may be biased comparison as ForestQC removes indels with high genotype missing rate in its filtering step. Consistent with the results of SNVs, the sample-level statistics indicate that each individual has fewer good indels from ForestQC than those from VQSR and ABHet (Figure 5 d, Supplementary Figure 10 c, d). Among good indels, ForestQC has 6% and 1% more novel indels than VQSR and ABHet, respectively (Table 3). In terms of allele frequency, rare indels detected by ForestQC accounts for 12.35% and 3.49% larger proportions than those by VQSR and ABHet, respectively (Supplementary Table 5). Similar to the results of the BP dataset, we also observe that the missing rate of rare indels is lower than that of common indels (Supplementary Figure 11 c).

Feature importance in random forest classifier

ForestQC uses several sequencing features in the random forest classifier to predict whether a variant with undermined quality is good or bad. To understand which sequencing features are more important indicators for quality of variants than other features, we analyze weight or importance of each feature that the random forest classifier learns during its model training. We first find that GC-content has the lowest importance in both BP and PSP datasets and also for both SNVs and indels (Supplementary Figure 12). This means that GC-content may not be as a strong indicator of quality of variants as other features related to sequencing quality such as depth (DP) and genotype quality (GQ). Second, the results show that classification results are not determined by one or two most important features as there is no feature with much higher importance than other features except GC-content. This suggests that all sequencing features except GC-content are important indicators for quality of variants and need to be included in our model. We also check correlation among features and find that while certain pairs of features are highly correlated, some features have moderate or low correlation to other features, suggesting that they may capture different information on quality of genetic variants (Supplementary Figure 13). Third, we observe that the same features have different importance between the BP dataset and the PSP dataset. For example, for SNVs, an outlier ratio of GQ feature has the highest importance for the PSP dataset while it has the third lowest importance for the BP dataset (Supplementary Figure 12 a). Also, the importance of features varies between SNVs and indels. One example is a SD of DP feature that has the highest importance for SNVs in the BP dataset, but it has the third lowest importance for indels (Supplementary Figure 12 a, b). Therefore, these results suggest that each feature may have a different contribution to classification results depending on sequencing data and types of genetic variants.

Performance of VQSR with different settings

For SNVs, GATK recommends three SNV call sets for training its VQSR model; 1) SNVs found in HapMap ("HapMap"), 2) SNVs in the omni genotyping array ("Omni"), and 3) SNVs in the 1000 Genomes Project ("1000G"). According to the VQSR parameter recommendation, SNVs in HapMap and Omni call sets are considered to contain only true variants while SNVs in 1000G contain both true and false positive variants[36]. We call this recommended parameter setting "original VQSR." We, however, find that considering SNVs in Omni to contain both true and false positive variants considerably improves the quality of good SNVs from VQSR for the BP dataset. We call this modified parameter setting "Omni_Modified VQSR". Results show that the mean Ti/Tv on good novel SNVs from Omni_Modified VQSR is 1.79, which is much higher than that from original VQSR (1.42) and slightly higher than that from ForestQC (1.68) (Supplementary Figure 14 a). We also find that the mean number of total SNVs from Omni_Modified VQSR is 3.65M which is much smaller than that from original VQSR (3.95M) but higher than that from ForestQC (3.58M) (Supplementary Figure 14 b). In terms of other statistics, good SNVs from original VQSR has 4.14x higher ME rate, 8.96x higher genotype missing rate, and 1.16x higher genotype discordance rate than those SNVs from Omni_Modified VQSR (Supplementary Figure 14 c-e). Interestingly, we do not observe the improved performance of Omni_Modified VQSR for the PSP dataset as the mean novel Ti/Tv on good novel SNVs of Omni_Modified VQSR is 1.23, which is slightly smaller than that of original VQSR (1.27) (Supplementary Figure 14 a), although individuals have fewer good SNVs from Omni_Modified VQSR (3.53M) than that from original VQSR (3.75M) (Supplementary Figure 14 b). These results suggest that the performance of VQSR may change significantly depending on whether to consider a certain SNV call set to contain only true variants or both true and false positive variants, and it appears that the difference in performance is more noticeable in certain sequencing datasets than others.

Although Omni_Modified VQSR has slightly better Ti/Tv on good novel SNVs and identifies more good SNVs than does ForestQC, good SNVs from Omni_Modified VQSR have 2.32x higher ME rate, 10.71x higher genotype missing rate, and 1.33x higher genotype discordance rate than good SNVs from ForestQC (Supplementary Figure 14 c-e). Hence, the results show that good SNVs from ForestQC have higher quality than those from VQSR even with the modification in the parameter setting.

Discussion

We developed an accurate and efficient method called ForestQC to identify a set of variants with high sequencing quality from NGS data. ForestQC combines the traditional filtering approach for performing QC in GWAS and the classification approach that uses a machine learning algorithm to classify whether a variant has good quality. Our method first uses stringent filters to identify good and bad variants that unequivocally have high and low sequencing quality, respectively. ForestQC then trains a random forest classifier using the good and bad variants obtained from the filtering step, and predicts whether a variant with ambiguous quality (a gray variant) is good or bad in an unbiased manner. We applied our method to two WGS datasets where one dataset consists of related individuals from families and the other dataset has unrelated individuals. We demonstrated that good variants identified from ForestQC in both datasets had higher sequencing quality than those from other approaches such as VQSR and a filtering approach based on ABHet.

A main advantage of our approach over the traditional filtering approach is that our method does not attempt to classify gray variants using filters. It is difficult to determine the quality of those gray variants using filters if their QC metrics (e.g. genotype missing rate) are close to the thresholds of filters. Hence, ForestQC avoids a limitation of the traditional filtering approaches that determine the quality of every variant using filters, which may exclude some of good variants from the downstream analysis. We did not compare our approach with the traditional filtering approach used in GWAS that removes variants according to HWE p-values, ME rates and genotype missing rates. One main reason is that the performance of this approach changes dramatically depending on filters and thresholds for each filter, and there are numerous different thresholds of filters as well as many combinations of filters that could be tested. Another reason is that its performance could be arbitrarily determined depending on the filters we use. For example, if one filter is to remove any variants having more than zero Mendel errors, the ME rate of good variants would be zero, but we may be removing many other good variants. We checked the accuracy of a filtering approach based on ABHet as ABHet is often used in performing QC of NGS data and is a good indicator for variant quality [26, 40, 41]. Also, as this approach is not based on standard QC metrics such as genotype missing rate, its performance is independent of those metrics unlike the standard filtering approaches. We showed that our approach outperformed the ABHet approach as the quality of good variants from ForestQC was better than that from ABHet, regardless of similar total number of good variants, as demonstrated by ME rate, missing rate, genotype discordance rate and Ti/Tv ratio in the BP and PSP dataset.

Although our approach is similar to VQSR as both approaches train machine learning classifiers to predict quality of variants, they have a few distinct differences. First, our approach trains the model using good and bad variants detected from sequencing data on which quality control is performed, while VQSR uses variants in existing databases, such as HapMap and 1000 genomes, as its training set. As VQSR uses previously known variants for model training, good variants from VQSR are likely to contain more known (and likely to be common) variants than novel (and rare) variants. We showed in both WGS datasets that it did identify more common and known SNVs and indels as good variants than ForestQC. This may not be a desirable outcome for some sequencing studies if one of their main goals is to identify rare and novel variants not captured in chips. Another difference between ForestQC and VQSR is the set of features used in the classifiers. While both methods use features related to sequencing depth and genotyping quality, VQSR uses some features that are specifically calculated by GATK software while our method uses quality information reported in the standard VCF file. This suggests that our method is more generalizable than VQSR as it can be applied to VCF files generated from variant callers other than GATK. The last difference is the machine learning algorithms that ForestQC and VQSR use. Our method trains a random forest model while VQSR trains a Gaussian Mixture model. Using the BP dataset, we found that random forest model was more accurate and much faster than Gaussian Mixture model (Table 1, Supplementary Table 6).

In addition to SNVs, we applied our method to indels in both WGS datasets and found that indels had much lower sequencing quality than do SNVs as the fraction of good indels detected by ForestQC was considerably smaller than that of SNVs. This is somewhat expected because indel or structural variant calling is much more difficult than SNV calling from sequencing data, and some of them are likely to be false positives [42, 43]. It is, however, important to note that VQSR classifies many more indels as good variants than does ForestQC or ABHet, but those good indels from VQSR may not have high sequencing quality. We showed that good indels from VQSR had similar Mendelian error rate to that without performing QC, indicating the poor performance of VQSR on indels. VQSR considers indels from Mills gold standard call set[38] as true variants, and while those indels might represent true variant sites, it does not necessarily mean that genotyping on those sites is accurate. Therefore, genetic studies need to perform stringent QC on indels to remove those erroneous calls and not to have false positive findings in their downstream analysis.

We found that the performance of VQSR was improved dramatically for the BP dataset when we considered SNVs in Omni genotyping array to have both true and false positive sites, compared with when they were assumed to have all true sites. We, however, did not observe this performance enhancement for the PSP dataset. This suggests that users may need to try different parameter settings to obtain optimal results from VQSR for specific sequencing datasets they analyze. Another issue with VQSR and also with ABHet is that some of good SNVs or indels have high genotype missing rate and ME rate, which may not be suitable for the downstream analysis such as association analysis. Thus, those variants need to be filtered out separately, which means users may need to perform an additional filtering step in addition to applying VQSR and ABHet to the dataset. As the filtering step is incorporated in ForestQC, our method does not have this issue.

Our approach is an extension of a previous approach that uses a logistic regression model to predict the quality of variants in the BP dataset[31]. While our approach is similar to the previous approach in that they both combine filtering and classification approaches, ForestQC uses a random forest classifier that has higher accuracy than a logistic regression model, according to our simulation results. It includes more bad variants for model training, leading to predictions with fewer biases. ForestQC also includes more features than the previous approach as well as more filters to improve the quality of good variants. Additionally, compared with the previous approach, ForestQC is more user-friendly and generalizable because users can choose or define different features and filters and tune the parameters according to their research goals.

ForestQC is efficient, modularized and flexible with following features. First, users are allowed to change thresholds for filters as needed. This is important because filters that are stringent for one dataset may not be stringent for another dataset. For example, variants from sequence data with very small sample size (e.g. < 100) may not have statistical power to have significant HWE p-values, and hence higher p-value thresholds may need to be used, compared with studies with larger sample size. If filters are not stringent enough, there may be many bad variants, and ForestQC would train a very stringent classifier, leading to the possible removal of good variants. On the contrary, if the filters are too stringent, there would be too few good variants or bad variants, which would lower the accuracy of our random forest classifier. In this study, after the filtering step, 4.39% of SNVs and 15.72% of indels in the BP dataset, and 5.06% of SNVs and 15.66% of indels in PSP dataset, were determined as bad variants. Empirically, we suggest filters for ForestQC such that after the filtering step, a fraction of bad variants is about 4-16%. Second, users are allowed to use their own filters and features provided that they specify values for those new filters and features at each variant site, and our software also allows users to remove existing filters and features. As there may be filters and features that capture sequencing quality of variants more accurately than current set of filters and features, this option allows users to improve ForestQC further. Third, ForestQC generates the probability of each gray variant being a good variant. This probability needs to be greater than a certain threshold for a gray variant to be predicted to be good, and it can also be used to analyze sequencing quality of certain variants. If studies find that a certain gray variant is associated with a phenotype, they may consider checking whether its probability of being a good variant is high enough. Lastly, ForestQC allows users to change the probability threshold for determining whether each gray variant is good or bad. Users may lower this threshold if they are interested in obtaining more good variants at the cost of including more bad variants.

Methods

ForestQC

ForestQC consists of two approaches: a filtering approach and a machine learning approach based on a random forest algorithm.

Filtering Given a variant call set from next generation sequencing data, ForestQC first applies several stringent filters to identify good, bad, and gray variants. Good variants are ones that pass all filters while bad variants fail any of them (Supplementary Table 1, 2). Gray variants are variants that neither pass filters for good variants nor fail filters for bad variants. We use following filters in the filtering step.

- Mendelian error (ME) rate. The Mendelian error occurs when a child's genotype is inconsistent with genotypes from parents. ME rate is calculated as the number of ME among all trios divided by the number of trios for a given variant. Note that this statistic is only available for family-based data.
- Genotype missing rate. This is the proportion of missing alleles in each variant.
- Hardy-Weinberg equilibrium (HWE) p-value. This is a p-value for hypothesis testing whether a variant is in Hardy-Weinberg equilibrium. Its null hypothesis is that the variant is in Hardy-Weinberg equilibrium. We use the algorithm used in an open-source software, VCFtools[44] for the calculation of Hardy-Weinberg equilibrium p-value.
- ABHet. This is allele balance for heterozygous calls. ABHet is calculated as the number of reference reads from individuals with heterozygous genotypes divided by the total number of reads from such individuals, which is supposed to be 0.50 for good variants. For variants in chromosome X, we only calculate ABHet for females.

Random forest classifier Random forest algorithm is a machine learning algorithm that runs efficiently on large datasets with high accuracy[35]. Briefly, random forest builds several randomized decision trees, each of which is trained to classify the input objects. For classification of a new object, the fitted random forest model passes the input vector down to each of the decision trees in the forest. Each decision tree has its classification result, then the forest would output the classification that the majority of the decision trees make. Balancing efficiency and accuracy, we train a random forest classifier using 50 decision trees (Supplementary Figure 1) and 50% as probability threshold (Supplementary Figure 2).

To train random forest, we use good and bad variants identified from the previous filtering step as a training dataset, after balancing their sample size by random sampling. Normally, good variants are much more numerous than bad variants, so we randomly sample from good variants with the sample size of bad variants. Hence, the sample size of the balanced training set would be twice as large as the sample size of bad variants. We also need features in training random forest, which characterize datasets, and we use following features.

- Mean and standard deviation of depth (DP) and genotyping quality (GQ). Depth and genotyping quality values are extracted from DP and GQ fields of each sample in VCF files, respectively, and mean and standard deviation are calculated over all samples for each variant.
- Outlier depth and outlier genotype quality. These are the proportions of samples whose DP or GQ is lower than a particular threshold. We choose this threshold as the first quartile value of all DP or GQ values of variants on chromosome 1. We use DP and GQ of variants on only chromosome 1 to reduce the computational costs.
- GC content: We first split a reference genome into window size of 1,000 bp and calculate GC content for each window as (# of G or C alleles) / (# of A, G, C or T alleles). Then, each variant is assigned a GC content value according to its position in the reference genome.

After training random forest with the training dataset using above features, we next use the fitted model to make predictions on gray variants on being good variants. Gray variants with the predicted probability of being good larger than 50% are labeled as predicted good variants. Then the predicted good variants and good variants from the previous filtering step are combined to form the final set of good variants. We apply the same procedure to identify bad variants.

Comparison of different machine learning algorithms

We compare nine different machine learning algorithms, in order to identify the best algorithm used for ForestQC. They are 1) k-nearest neighbors for supervised 2-class classification (8 threads); 2) logistic regression (8 threads); 3) support vector machine with Gaussian kernel function and penalty parameter C of 1 (1 thread); 4) random forest with 50 trees (8 threads); 5) naïve Bayes without any prior probabilities of the classes (1 thread); 6) artificial neural network with sigmoid function as activation function (8 threads); 7) AdaBoost with 50 estimators and learning rate of 1, which uses SAMME.R real boosting algorithm (1 thread); 8) quadratic discriminant analysis without any prior on classes. Its regularization is 0 and its threshold for rank estimation is 1e-4 (1 thread); and 9) Gaussian mixture assuming that each component has its own diagonal covariance matrix (1 thread). Other parameters of these machine learning algorithm are default, as described in the documentations of Python scikit-learn package [45]. All learning algorithms use the seven aforementioned features: mean and standard deviation of sequencing depth, mean and standard deviation of genotype quality, outlier depth, outlier quality and GC content.

To test these nine machine learning algorithms, we obtain training and test datasets from the BP dataset, using filters described in Supplementary Table 1 and 2. There are 21,248,103 good SNVs and 2,257,506 good indels while there are 1,100,325 bad SNVs and 624,965 bad indels. We sample 100,000 variants randomly from good variants and 100,000 variants from bad variants to generate a training set. Similarly, 100,000 good variants and 100,000 bad variants are randomly chosen from the rest of variants to form a test set. Each machine learning model shares the same training and test sets. We train the machine learning models and measure training time at a training stage, and then test their accuracy and measure prediction time at a testing stage. We measure both CPU time and real time of each algorithm where CPU time is the sum of all CPU time over all threads and real time measures the elapsed clock time between the start and end of each algorithm. To assess the performance of each algorithm, we compute F1-score for the test set. F1-score is the harmonic average of precision and recall, which is calculated as $2 \cdot precision \cdot recall / (precision + recall)$. The closer F1-score is to 1, the higher classification accuracy is. Recall is the fraction of true positive results over all samples that should be given positive prediction. Precision is the number of true positive results divided by the number of positive results predicted by the classifier.

ABHet approach and VQSR

We compare ForestQC with two other approaches for performing QC on genetic variants. One is a filtering approach based on ABHet and the other is a classification approach called VQSR from GATK software. For the ABHet approach, we consider variants with ABHet > 0.7 or < 0.3 as bad variants, and the rest as good variants. We chose this threshold setting of ABHet (> 0.3 and < 0.7) because the ADSP project could not reliably confirm heterozygous calls with ABHet > 0.7 with Sanger sequencing[26]. We also exclude variants with small ABHet values (< 0.3) to ensure high quality. For GATK, we use recommended arguments as of 2018-04-04[36]. For SNVs, VQSR takes SNVs in HapMap 3 release 3, 1000 Genome Project and Omni genotyping array as training resources, and dbSNP135 as known sites resource. HapMap and Omni sites are considered as true sites, meaning that SNVs in these datasets are all true variants, while 1000 Genome Project sites are regarded as false sites, meaning that there could be both true and false-positive variants. The desired level of sensitivity of true sites is set to be 99.5%. In the BP dataset, we run VQSR version 3.5-0-g36282e4 with following annotations; quality by depth (QD), RMS mapping quality (MQ), mapping quality rank sum test (MQRankSum), read position rank sum test (ReadPosRankSum), fisher strand (FS), coverage (DP), strand odds ratio (SOR) and inbreeding coefficient (InbreedingCoeff) to evaluate the likelihood of true positive calls. In the PSP dataset, we use VQSR version 3.2-2-gec30cee that uses all previous annotations except SOR because variants in PSP dataset do not have the SOR annotation. For indels, VQSR takes indels in Mills gold standard call set[38] as true training resource, and dbSNP135 as known sites resource. The desired level of sensitivity of true sites is set to be 99.0%. We use VQSR version 3.5-0-g36282e4 with QD, DP, FS, SOR, ReadPosRankSum, MQRankSum and InbreedingCoeff annotations to evaluate the likelihood of true positive calls in the BP dataset, while we run VQSR version 3.2-2-gec30cee with the same annotations except SOR for the PSP dataset.

Performance metrics

21 sample-level metrics and 20 variant-level metrics are defined to measure the sequencing quality of the variant call set after performing quality control (Supplementary Table 7). Variant-level metrics provide us with a summarized assessment report of the sequencing quality of a variant call set, such as total SNVs of the whole dataset. They are calculated based on the information of all variants in a variant call set. For example, the number and the proportion of multi-allelic SNVs are counted for the entire dataset, each of which is identified according to its reference and alternate alleles. On the other hand, sample-level metrics enable the inspection of the sequencing quality for sequenced individuals in a variant call set. For instance, we check the distribution of novel Ti/Tv or other quality metrics among all individuals in the study. Sample-level metrics are calculated for each sample, using its genotype information on all variants in the dataset, and a distribution of those metrics across all individuals is shown as a box plot. For example, the number of SNV singletons on a sample level shows the distribution of the number of SNV singletons across all sequenced individuals. In this study, both sample-level and variant-level metrics are used to evaluate the sequencing quality of WGS variant datasets.

BP and PSP WGS datasets

The BP WGS dataset is for studying bipolar disorder whose average coverage is 36. This study recruited individuals from 11 Colombia (CO) and 15 Costa Rica (CR) extended pedigrees in total. 454 subjects from 10 CO and 12 CR families are both whole genome sequenced and genotyped with microarray. There are 144 individuals diagnosed with BP1 and 310 control samples that are unaffected or have non-BP traits. GATK-HaplotypeCaller 3.5-0-g36282e4 according to the GATK best practices[23] is used to call variants for the BP dataset and the reference genome is HG19. After initial QC on individuals, five individuals are removed because of poor sequencing quality and possible sample mix-ups. Finally, 449 individuals are included in an analysis, resulting in 25,081,636 SNVs and 3,976,710 indels. 1,814,326 SNVs in the WGS dataset are also genotyped with microarray, which are used to calculate genotype discordance rate. In this study, we use the BP dataset before any QC performed on genetic variants. In a previous study [31], genetic variants in the BP WGS dataset are first processed with VQSR and then filtered with a trained logistic regression model to remove variants with low quality.

The PSP WGS dataset is for studying progressive supranuclear palsy with average coverage of 29. 544 unrelated individuals are whole genome sequenced, 518 of whom are also genotyped with microarray. Among them, 119 individuals have 547,644 SNPs and 399 individuals have 1,682,489 SNPs genotyped with microarray, respectively. That 119 individuals would be excluded when calculating genotype discordance rate in case of biases caused by fewer SNPs. There are 356 individuals diagnosed with PSP and 188 individuals as controls. Variant calling for the PSP dataset is performed by GATK-HaplotypeCaller 3.2-2-gec30cee with the GATK best practices pipeline where the reference genome is HG19. 49 samples are found to have high missing rate or high relatedness with other samples, or are diagnosed with diseases other than PSP, so they are removed. Next, we extract variant data with only 495 individuals with VCFtools. Monomorphic variants are then removed. After preprocessing, the PSP WGS dataset has 33,273,111 SNVs and 5,093,443 indels. There are 1,682,489 SNVs from 381 samples genotyped by both microarray and WGS, which are used for calculating genotype discordance rate.

References

[1] Pray, L. Genome-wide association studies and human disease networks. *Nature Education* **1**, 220 (2008).

[2] Hirschhorn, J. N. & Daly, M. J. Genome-wide association studies for common diseases and complex traits. *Nat. Rev. Genet.* **6**, 95–108 (2005).

[3] Schizophrenia Working Group of the Psychiatric Genomics Consortium. Biological insights from 108 schizophrenia-associated genetic loci. *Nature* **511**, 421–427 (2014).

[4] Sladek, R. *et al.* A genome-wide association study identifies novel risk loci for type 2 diabetes. *Nature* **445**, 881–885 (2007).

[5] Ng, M. C. Y. *et al.* Meta-analysis of genome-wide association studies in african americans provides insights into the genetic architecture of type 2 diabetes. *PLoS Genet.* **10**, e1004517 (2014).

[6] Nalls, M. A. *et al.* Large-scale meta-analysis of genome-wide association data identifies six new risk loci for parkinson's disease. *Nat. Genet.* **46**, 989–993 (2014).

[7] Visscher, P. M., Brown, M. A., McCarthy, M. I. & Yang, J. Five years of GWAS discovery. *Am. J. Hum. Genet.* **90**, 7–24 (2012).

[8] Goldstein, D. B. Common genetic variation and human traits. *N. Engl. J. Med.* **360**, 1696–1698 (2009).

[9] Cirulli, E. T. & Goldstein, D. B. Uncovering the roles of rare variants in common disease through whole-genome sequencing. *Nat. Rev. Genet.* **11**, 415–425 (2010).

[10] Bodmer, W. & Bonilla, C. Common and rare variants in multifactorial susceptibility to common diseases. *Nat. Genet.* **40**, 695–701 (2008).

[11] Schork, N. J., Murray, S. S., Frazer, K. A. & Topol, E. J. Common vs. rare allele hypotheses for complex diseases. *Curr. Opin. Genet. Dev.* **19**, 212–219 (2009).

[12] Pritchard, J. K. Are rare variants responsible for susceptibility to complex diseases? *Am. J. Hum. Genet.* **69**, 124–137 (2001).

[13] Rivas, M. A. *et al.* Deep resequencing of GWAS loci identifies independent rare variants associated with inflammatory bowel disease. *Nat. Genet.* **43**, 1066–1073 (2011).

[14] Gudmundsson, J. *et al.* A study based on whole-genome sequencing yields a rare variant at 8q24 associated with prostate cancer. *Nat. Genet.* **44**, 1326–1329 (2012).

[15] Lange, L. A. *et al.* Whole-exome sequencing identifies rare and low-frequency coding variants associated with LDL cholesterol. *Am. J. Hum. Genet.* **94**, 233–245 (2014).

[16] Yu, T. W. *et al.* Using whole-exome sequencing to identify inherited causes of autism. *Neuron* **77**, 259–273 (2013).

[17] Dohm, J. C., Lottaz, C., Borodina, T. & Himmelbauer, H. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.* **36**, e105 (2008).

[18] Nakamura, K. *et al.* Sequence-specific error profile of illumina sequencers. *Nucleic Acids Res.* **39**, e90 (2011).

[19] Ross, M. G. *et al.* Characterizing and measuring bias in sequence data. *Genome Biol.* **14**, R51 (2013).

[20] Schirmer, M., D'Amore, R., Ijaz, U. Z., Hall, N. & Quince, C. Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. *BMC Bioinformatics* **17**, 125 (2016).

[21] Manley, L. J., Ma, D. & Levine, S. S. Monitoring error rates in illumina sequencing. *J. Biomol. Tech.* **27**, 125–128 (2016).

[22] Poptsova, Maria S., Irina A. Il'icheva, Dmitry Yu Nechipurenko, Larisa A. Panchenko, Mingian V. Khodikov, Nina Y. Oparina, Robert V. Polozov, Yury D. Nechipurenko, and Sergei L. Grokhovsky. Non-Random DNA Fragmentation in next-Generation Sequencing. *Scientific Reports* **4**, 4532 (2014).

[23] DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics* **43**, 491–498 (2011).

[24] Highnam, G. *et al.* An analytical framework for optimizing variant discovery from personal genomes. *Nat. Commun.* **6**, 6275 (2015).

[25] O'Rawe, J. *et al.* Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome Med.* **5**, 28 (2013).

[26] ADSP. Review and Proposed Actions for False-Positive Association Results in ADSP Case-Control Data | ADSP. (2016). Available at: https://www.niagads.org/adsp/content/review-and-proposed-actions-false-positive-association-results-adsp-case-control-data.

[27] McKenna, A. *et al.* The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research* **20**, 1297–1303 (2010).

[28] Guo, Y. *et al.* Multi-perspective quality control of illumina exome sequencing data using QC3. *Genomics* **103**, 323–328 (2014).

[29] 1000 Genomes Project Consortium *et al.* A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).

[30] International HapMap Consortium. The international HapMap project. *Nature* **426**, 789–796 (2003).

[31] Sul, J. H. *et al.* Contribution of common and rare variants to bipolar disorder susceptibility in extended pedigrees from population isolates. *bioRxiv* 363267 (2018). doi:10.1101/363267

[32] Clark, MJ. *et al*. Performance comparison of exome DNA sequencing technologies. *Nat. Biotechnol.* **29**, 908-914 (2011)

[33] Wang W. *et al.* Next generation sequencing has lower sequence coverage and poorer SNP-detection capability in the regulatory regions. *Sci. Rep.* **1** 55 (2011)

[34] Aird, D. *et al.* Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.* **12**, R18 (2011).

[35] Breiman, L. Random forests. *Machine learning* **45**, 5–32 (2001).

[36] GATK Dev Team. Which training sets / arguments should I use for running VQSR? (2017). Available at: https://software.broadinstitute.org/gatk/documentation/article.php?id=1259.

[37] Bainbridge, M. N. *et al.* Targeted enrichment beyond the consensus coding DNA sequence exome reveals exons with higher variant densities. *Genome Biol.* **12**, R68 (2011).

[38] Mills, R. E. *et al.* Natural genetic variation caused by small insertions and deletions in the human genome. *Genome Res.* **21**, 830–839 (2011).

[39] Sudmant, P. H. *et al.* An integrated map of structural variation in 2,504 human genomes. *Nature* **526,** 75–81 (2015).

[40] Aylward, A. *et al.* Using Whole Exome Sequencing to Identify Candidate Genes With Rare Variants In Nonsyndromic Cleft Lip and Palate. *Genet. Epidemiol.* **40,** 432–441 (2016).

[41] Bellenguez, C. *et al.* Contribution to Alzheimer's disease risk of rare variants in TREM2, SORL1, and ABCA7 in 1779 cases and 1273 controls. *Neurobiol. Aging* **59,** 220.e1–220.e9 (2017).

[42] Tattini, L., D'Aurizio, R. & Magi, A. Detection of Genomic Structural Variants from Next-Generation Sequencing Data. *Front Bioeng Biotechnol* **3,** 92 (2015).

[43] Hasan, MS *et al*. Performance Evaluation of Indel Calling Tools Using Real Short-Read Data. *Human Genomics* **9**, 20 (2015).

[44] Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27**, 2156–2158 (2011).

[45] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[46] Kelly, B. J. *et al.* Churchill: an ultra-fast, deterministic, highly scalable and balanced parallelization strategy for the discovery of human genetic variation in clinical and population-scale genomics. *Genome Biol.* **16,** 6 (2015).

Acknowledgements

We thank Dr. Susan K. Service from Department of Psychiatry and Biobehavioral Sciences, UCLA for the precious comments and suggestions to our project and this manuscript. We thank all study participants in the BP and PSP datasets. This work was supported by NIMH grants R01 MH075007 and R01 MH095454, and NIEHS grant K01 ES028064.

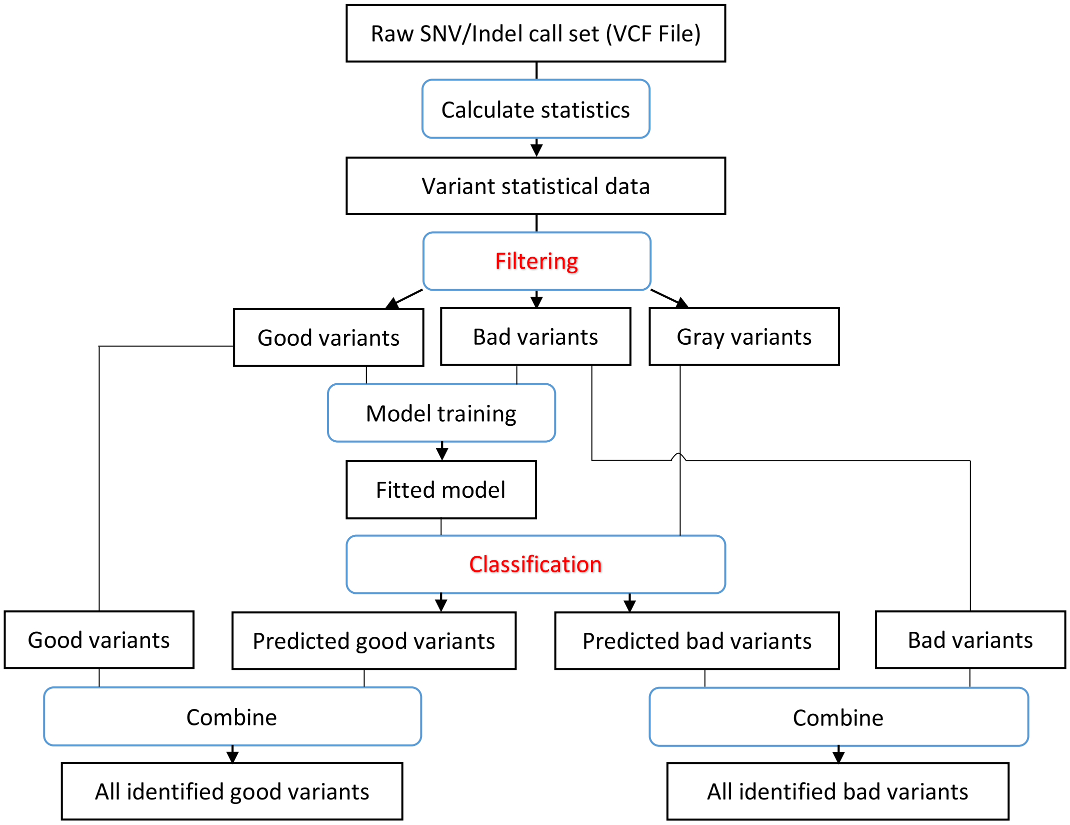


Figure 1: Workflow of ForestQC. ForestQC takes a raw variant call set in the VCF format as input. Then it calculates the statistics of each variants, including MAF, mean depth, mean genotyping quality, etc.. In the filtering step, it separates the variant call set into good, bad, and gray variants by applying various hard filters, such as Mendelian error rate and genotype missing rate. In classification step, good and bad variants are used to train a random forest model, which is then applied to assign labels to gray variants. Variants predicted to be good among gray variants are combined with good variants from the classification step for the final set of good variants. The same procedure applies to find the final set of bad variants.

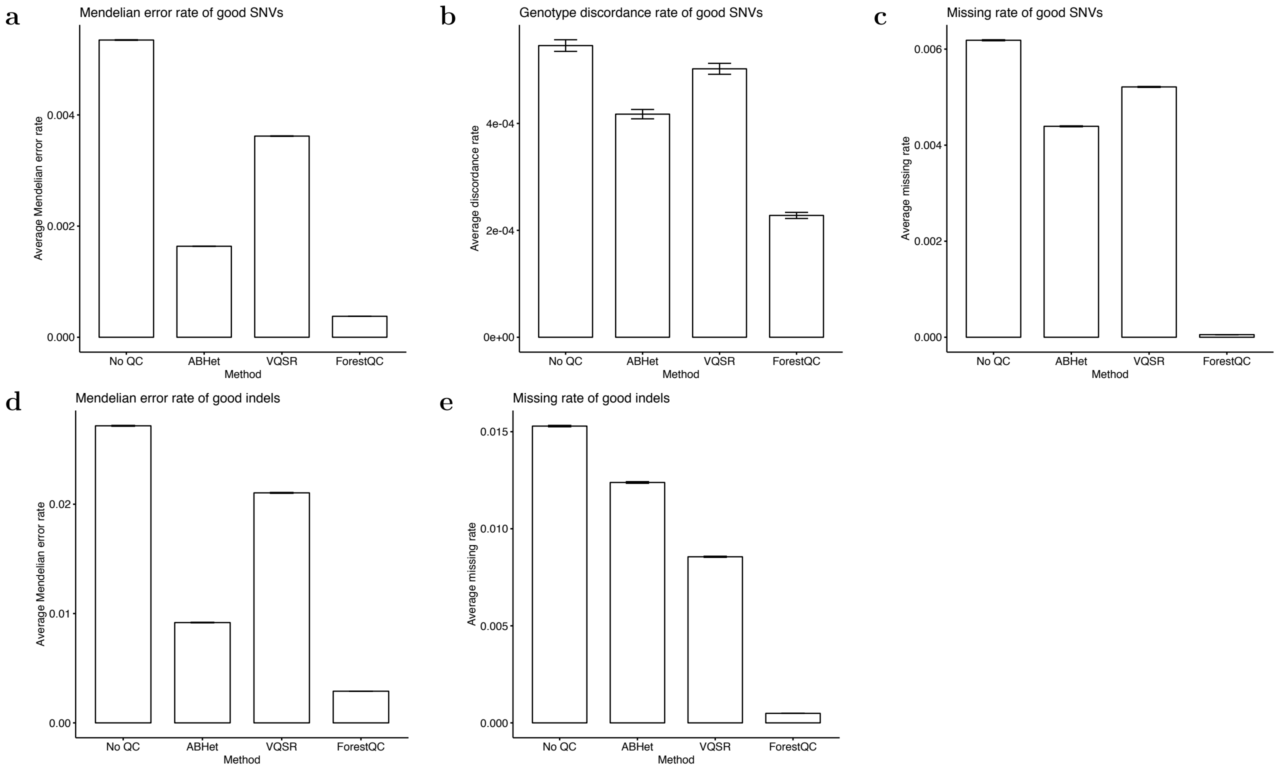


Figure 2: Overall quality of good variants in the BP dataset detected by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. The average Mendelian error rate and genotype missing rate for SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.

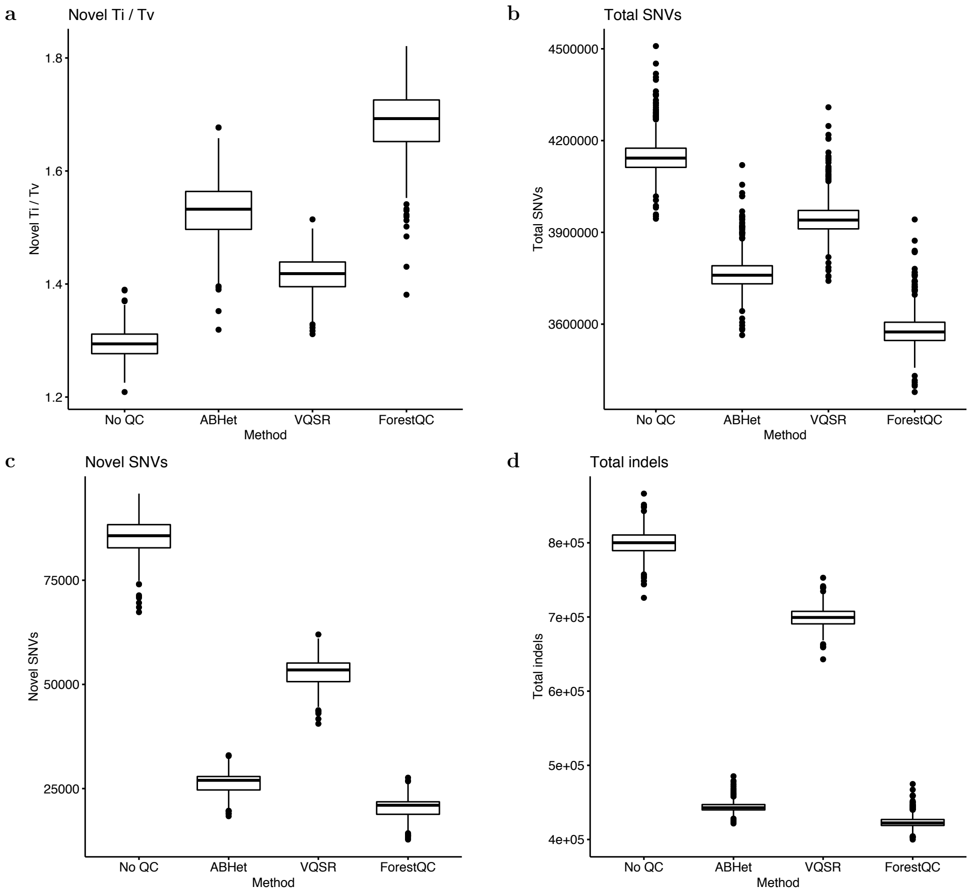


Figure 3: Sample-level quality metrics of good variants in the BP dataset identified by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. (a) Ti/Tv ratio of SNVs not found in dbSNP. (b) Total number of SNVs. (c) The number of SNVs not found in dbSNP. (d) Total number of indels. The version of dbSNP is 150.

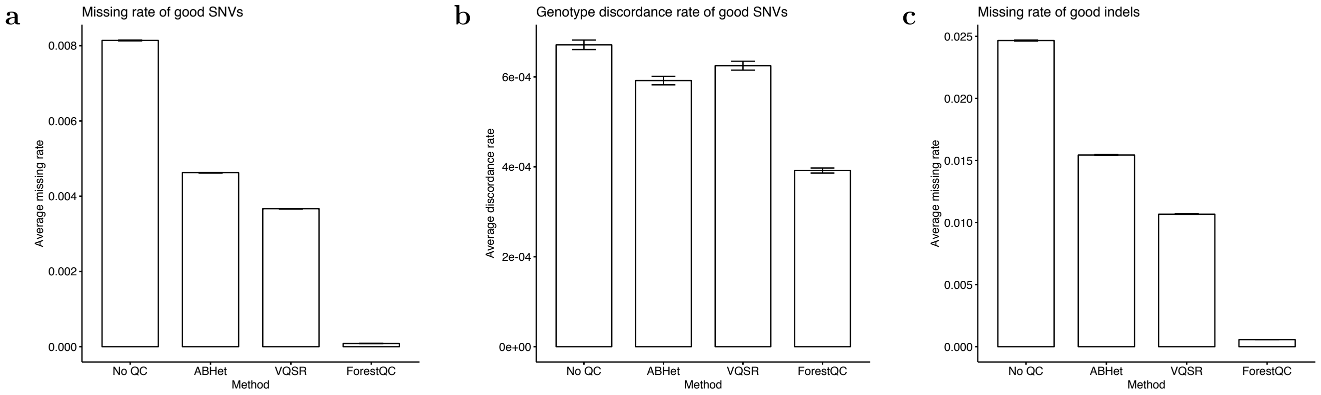


Figure 4: Overall quality of good variants in the PSP dataset detected by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. The average genotype missing rate for both SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.

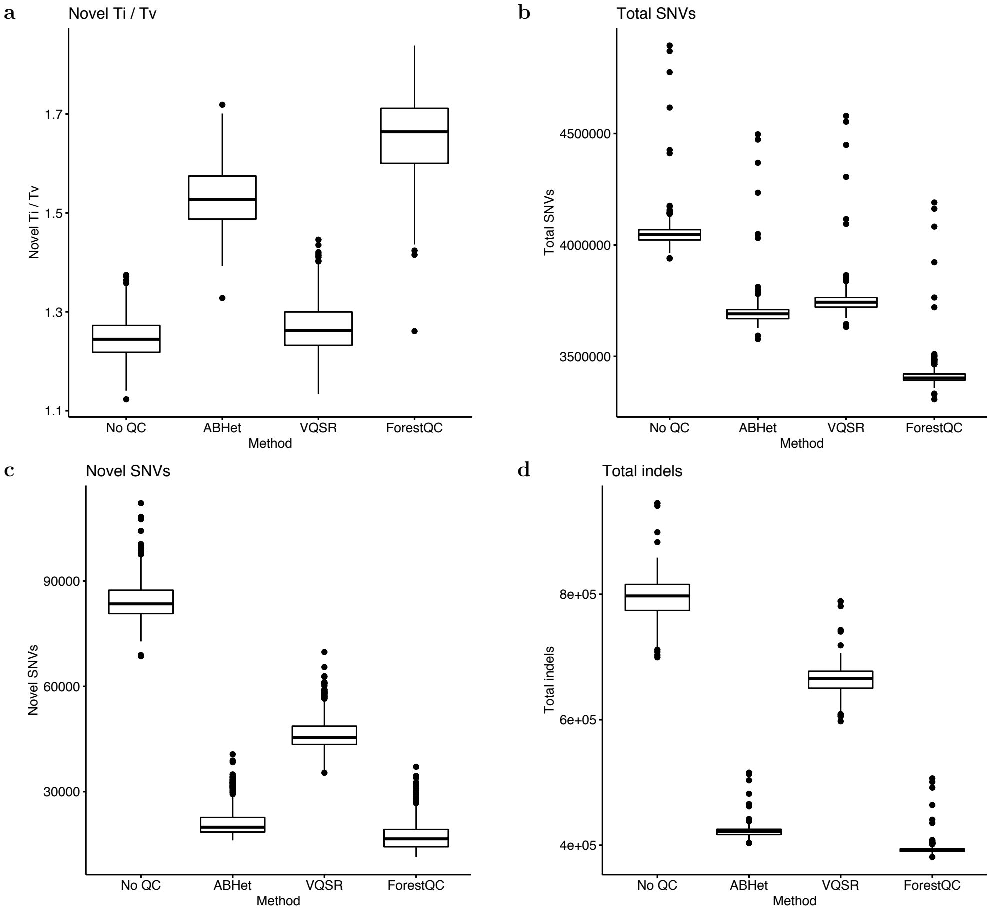


Figure 5: Sample-level quality metrics of good variants in the PSP dataset identified by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. (a) Ti/Tv ratio of SNVs not found in dbSNP. (b) Total number of SNVs. (c) The number of SNVs not found in dbSNP. (d) Total number of indels. The version of dbSNP is 150.

Machine learning algorithms	CPU time (sec)	Real time (sec)	F1-score for indel classification	F1-score for SNV classification
Random Forest	89.64	16.65	0.9428	0.9737
ANN	1210.54	221.36	0.9439	0.9725
SVM	1696.19	1698.15	0.9383	0.9702
AdaBoost	34.18	34.17	0.9288	0.9676
Logistics Regression	3.05	2.91	0.9080	0.9665
KNN	45.83	7.46	0.9197	0.9482
QDA	0.57	0.37	0.8998	0.9201
Native Bayes	0.22	0.23	0.8724	0.8964
Gaussian Mixture	54.49	45.57	0.8343	0.2688

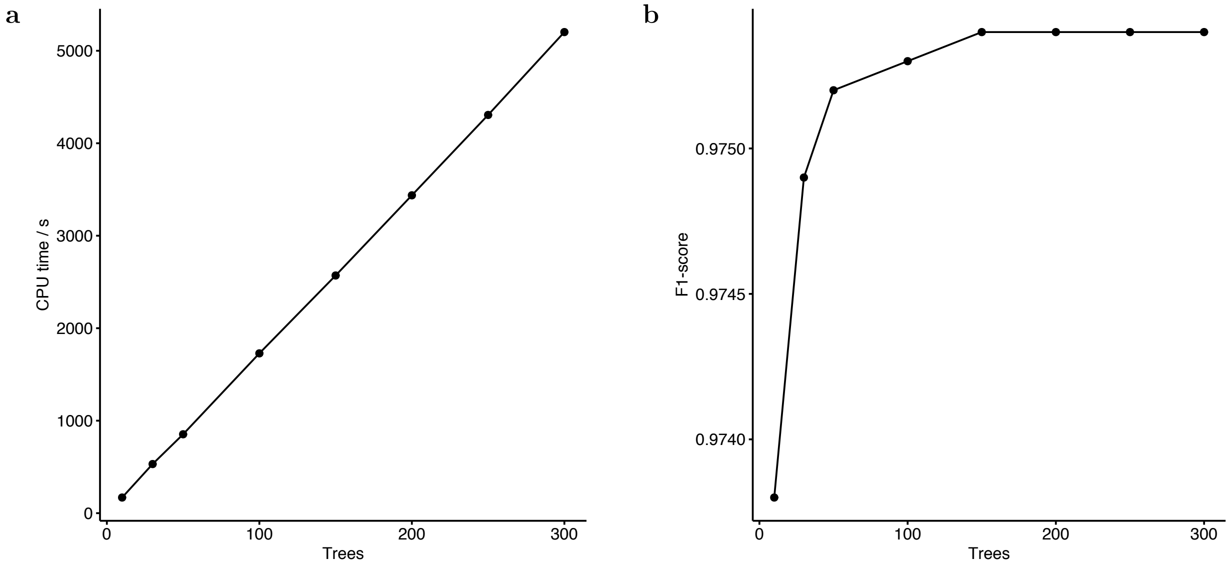
Table 1: Performance of nine different machine learning algorithms, including F1-score, total CPU time and real time cost of model fitting and prediction, ranked by F1-score for SNV classification. Random forest, ANN, logistic regression and KNN are set to run with eight threads. “ANN”: artificial neural network. “SVM”: support vector machine. “KNN”: K-nearest neighbors classifier. “QDA”: quadratic discriminant analysis.

Metric	No QC	ABHet	VQSR	ForestQC
Total SNVs	25,081,636	22,415,368	24,108,749	22,227,503
Known SNVs	21,165,051	19,665,276	20,593,216	19,361,635
Known SNVs (%)	84.38%	87.73%	85.42%	87.11%
Novel SNVs	3,916,585	2,750,092	3,515,533	2,865,868
Novel SNVs (%)	15.62%	12.27%	14.58%	12.89%
Known Ti / Tv	2.0751	2.1541	2.1212	2.1678
Novel Ti / Tv	1.5262	1.7987	1.6258	1.7790
Total indels	3,976,710	2,670,647	3,142,076	2,789,037
Known indels	3,094,271	2,188,996	2,682,074	2,237,002
Known indels (%)	77.81%	81.97%	85.36%	80.21%
Novel indels	882,439	481,651	460,002	552,035
Novel indels (%)	22.19%	18.03%	14.64%	19.79%
Multi-allelic SNVs	153,836	26,549	124,269	77,693
Multi-allelic SNVs (%)	0.61%	0.12%	0.52%	0.35%
Known multi-allelic SNVs	134,108	24,880	112,839	75,107
Known multi-allelic SNVs (%)	0.63%	0.13%	0.55%	0.39%
Singletons in SNVs	3,983,906	3,650,571	3,945,806	3,801,389
Singletons in SNVs (%)	15.88%	16.29%	16.37%	17.10%
Singletons in indels	485,453	395,798	365,633	433,222
Singletons in indels (%)	12.21%	14.82%	11.64%	15.53%

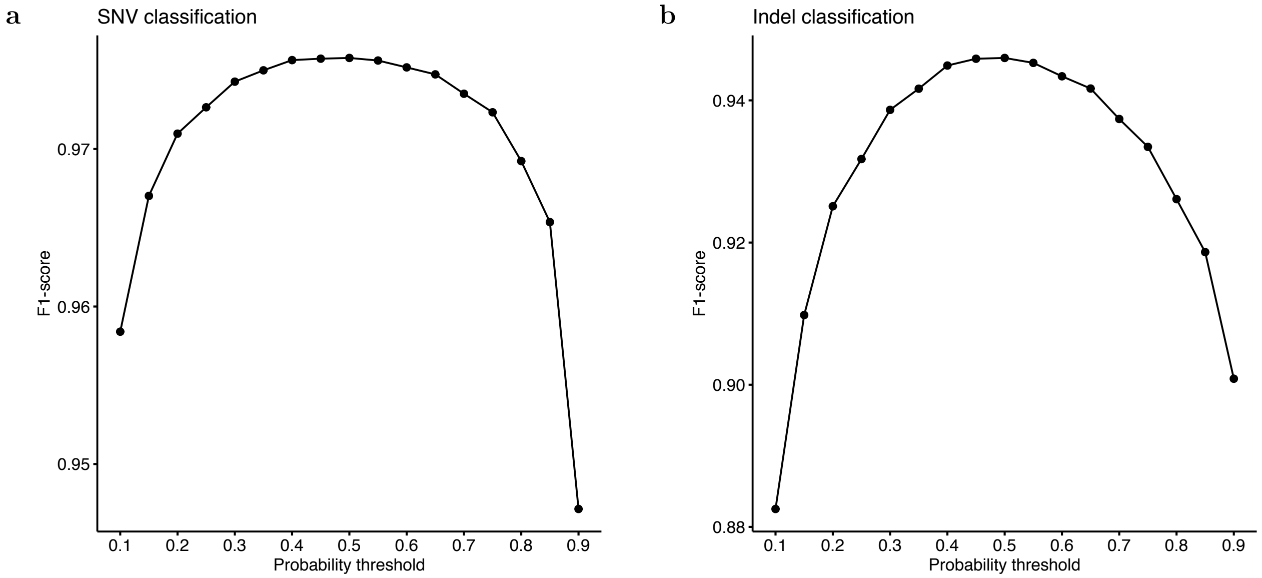
Table 2: Variant-level quality metrics of good variants in the BP dataset processed by different methods, including no QC applied, ABHet approach, VQSR and ForestQC. There are 20 metrics in total, which are described in Methods section in detail. “Known” stands for variants found in dbSNP. “Novel” stands for variants not found in dbSNP. The version of dbSNP is 150

Metric	No QC	ABHet	VQSR	ForestQC
Total SNVs	33,273,111	29,771,182	31,281,620	29,352,329
Known SNVs	25,960,464	24,142,744	24,910,728	23,514,257
Known SNVs (%)	78.02%	81.09%	79.63%	80.11%
Novel SNVs	7,312,647	5,628,438	6,370,892	5,838,072
Novel SNVs (%)	21.98%	18.91%	20.37%	19.89%
Known Ti / Tv	2.1202	2.1811	2.1643	2.1990
Novel Ti / Tv	1.6253	1.8236	1.7089	1.8131
Total indels	5,093,443	3,311,136	3,682,319	3,418,242
Known indels	3,679,990	2,532,899	3,012,662	2,567,879
Known indels (%)	72.25%	76.50%	81.81%	75.12%
Novel indels	1,413,453	778,237	669,657	850,363
Novel indels (%)	27.75%	23.50%	18.19%	24.88%
Multi-allelic SNVs	250,418	6,685	188,180	146,247
Multi-allelic SNVs (%)	0.75%	0.02%	0.60%	0.50%
Known multi-allelic SNVs	219,411	6,210	174,818	138,890
Known multi-allelic SNVs (%)	0.85%	0.03%	0.70%	0.59%
Singletons in SNVs	14,768,613	13,849,361	14,390,515	14,055,519
Singletons in SNVs (%)	44.39%	46.52%	46.00%	47.89%
Singletons in indels	1,582,090	1,350,433	1,242,934	1,393,736
Singletons in indels (%)	31.06%	40.78%	33.75%	40.77%

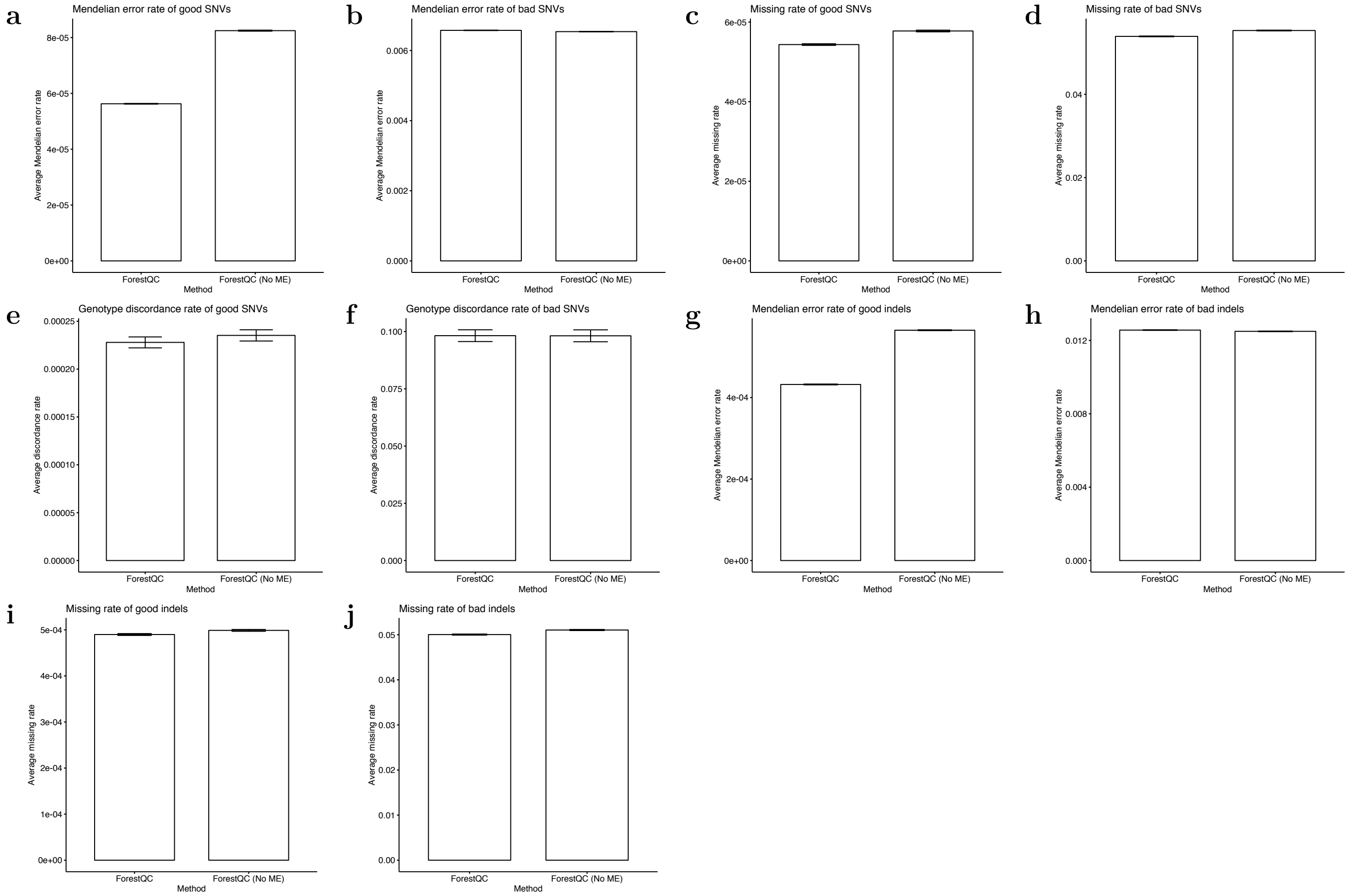
Table 3: Variant-level quality metrics of good variants in the PSP dataset processed by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. There are 20 metrics in total, which are described in Methods section in detail. “Known” stands for variants found in dbSNP. “Novel” stands for variants not found in dbSNP. The version of dbSNP is 150



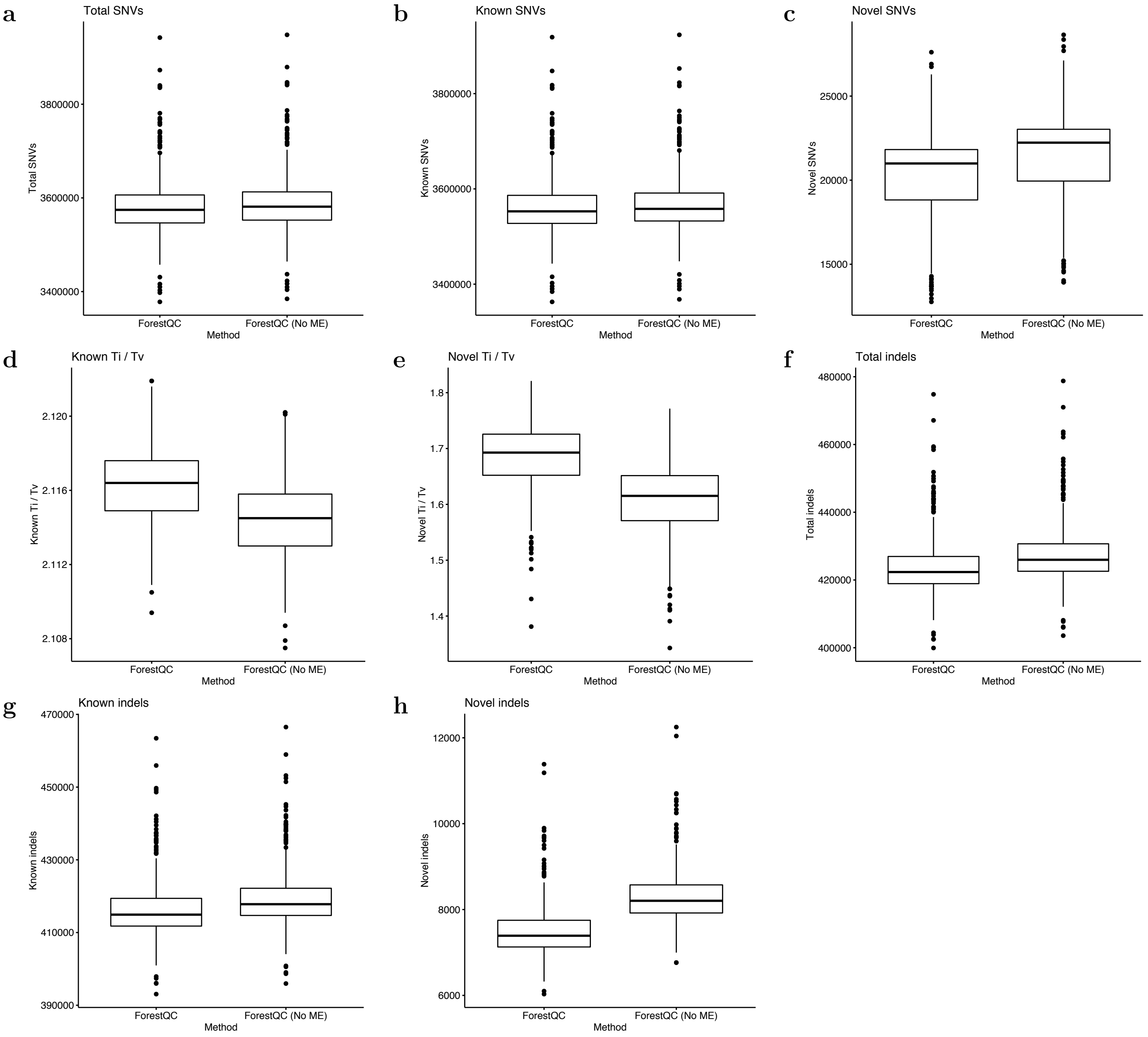
Supplementary Figure 1: Relationship between the number of trees in random forest model and the performance of ForestQC. Relationship between the number of trees and (a) CPU time and (b) F1-score.



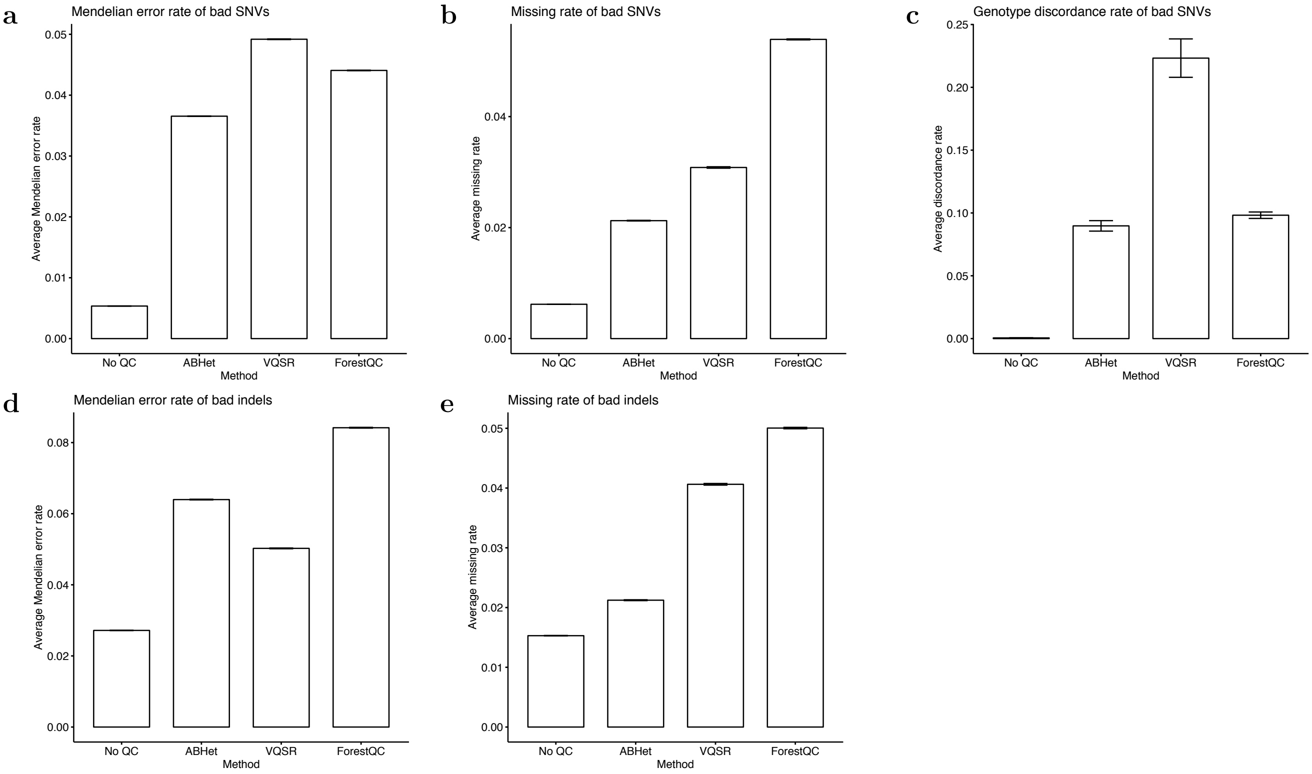
Supplementary Figure 2: Relationship between the probability threshold for predicting a variant to be good and the precision of ForestQC. If the probability of a variant predicted to be good is larger than the probability threshold, this variant would be labeled as a good variant. Classification precision changes along with the probability threshold in SNV classification (a) and indel classification (b). The precision of ForestQC is measured in F1-score.



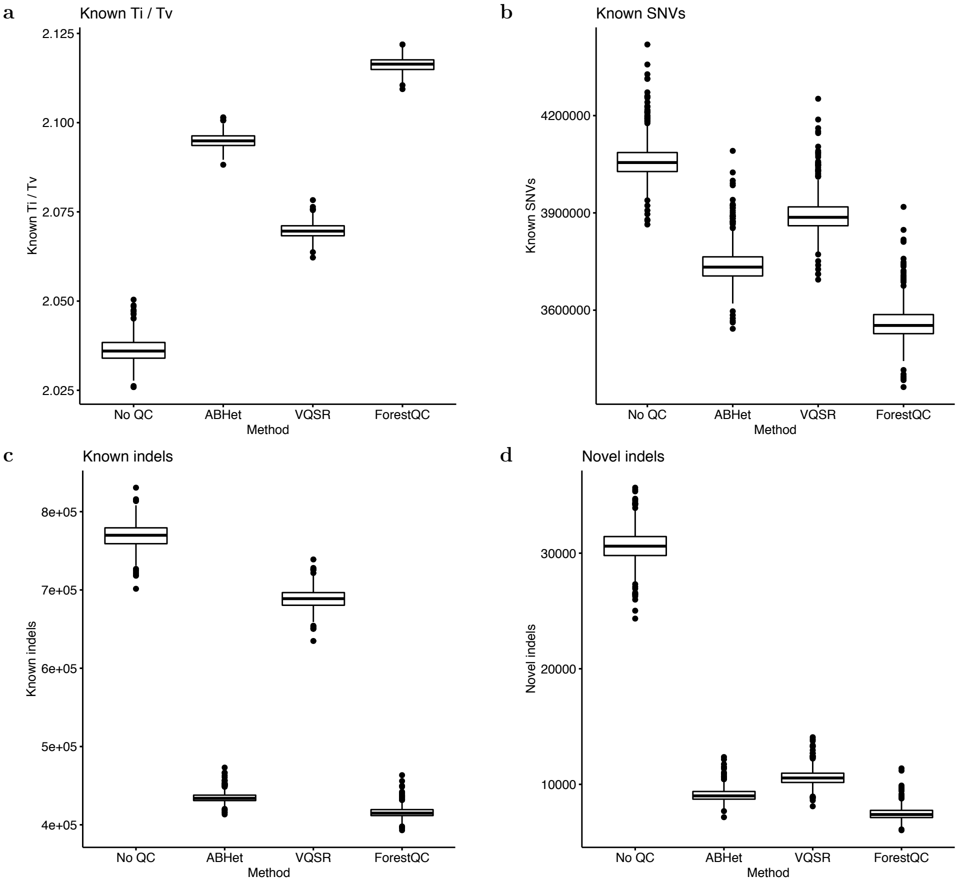
Supplementary Figure 3: Overall quality of good and bad variants in the BP dataset identified by ForestQC using ME rate as a filter or not. The average Mendelian error rate and genotype missing rate for SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.



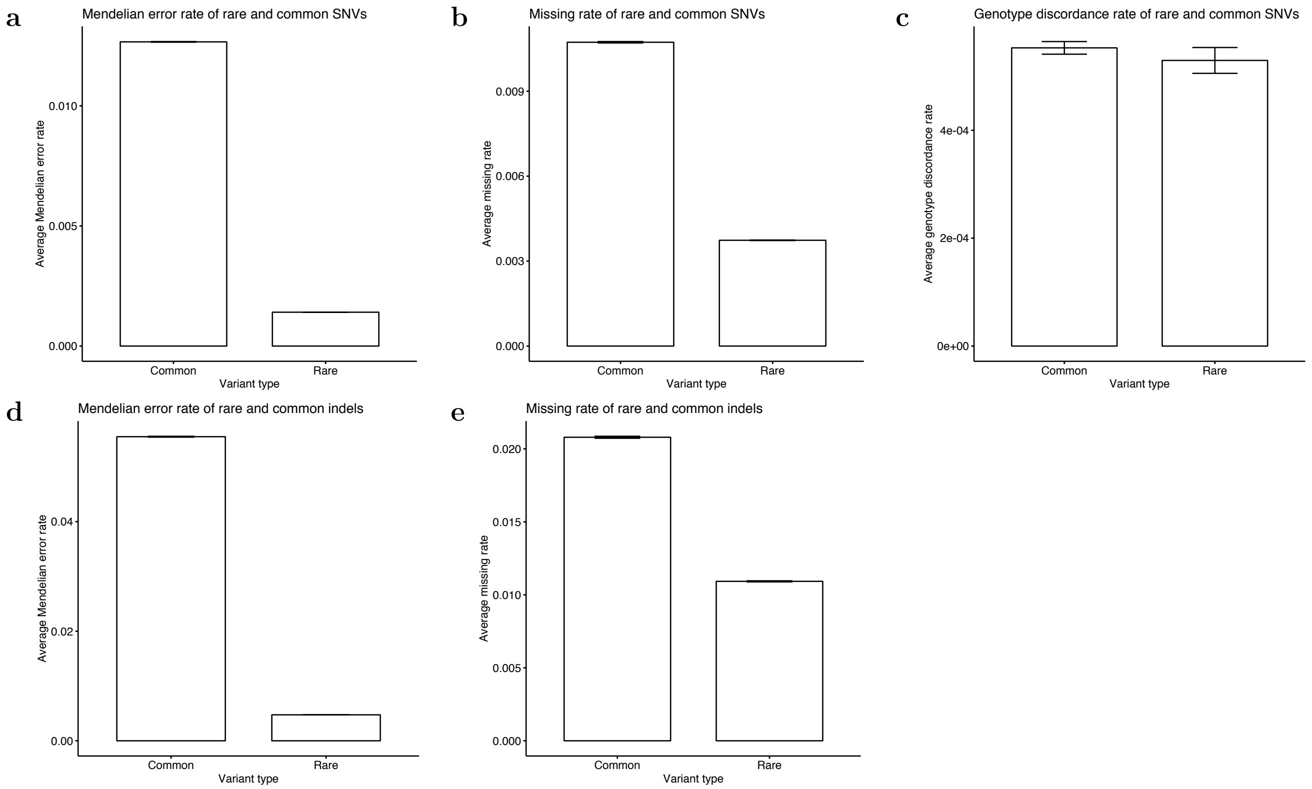
Supplementary Figure 4: Sample-level quality metrics of good variants in the BP dataset identified by ForestQC using ME rate as a filter or not. (a) Total number of SNVs. (b) The number of SNVs found in dbSNP. (c) the number of SNVs not found in dbSNP. (d) Ti/Tv ratio of SNVs found in dbSNP. (e) Ti/Tv ratio of SNVs not found in dbSNP. (f) Total number of indels. (g) the number of indels found in dbSNP. (h) the number of indels not found in dbSNP. The version of dbSNP is 150.



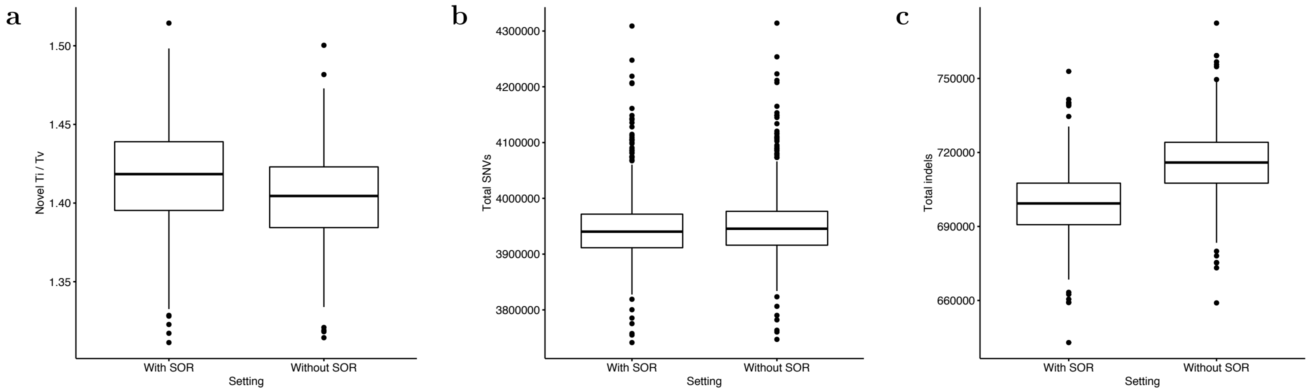
Supplementary Figure 5: Overall quality of bad variants in the BP dataset detected by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. The average Mendelian error rate and genotype missing rate for SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.



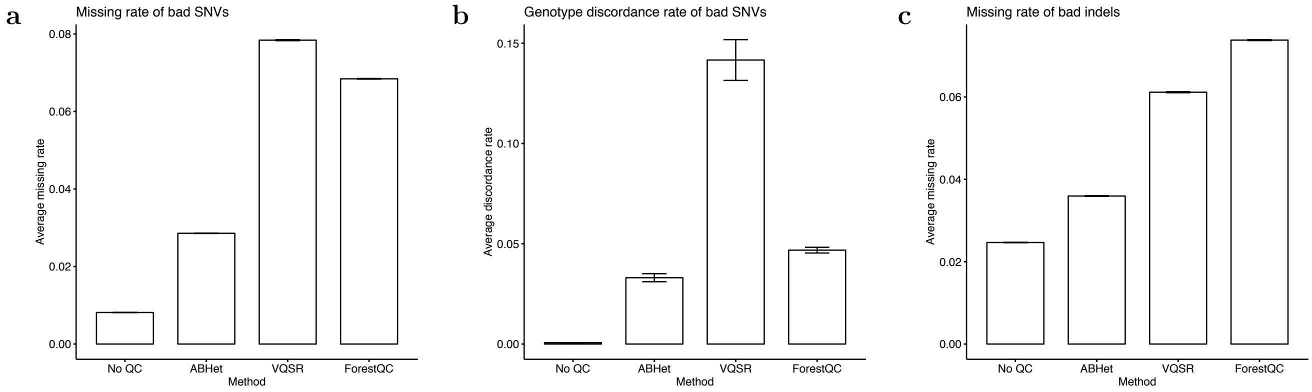
Supplementary Figure 6: Sample-level quality metrics of good variants in the BP dataset identified by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. (a) Ti/Tv ratio of SNVs found in dbSNP. (b) The number of SNVs found in dbSNP. (c) The number of indels found in dbSNP. (d) The number of indels not found in dbSNP. The version of dbSNP is 150.



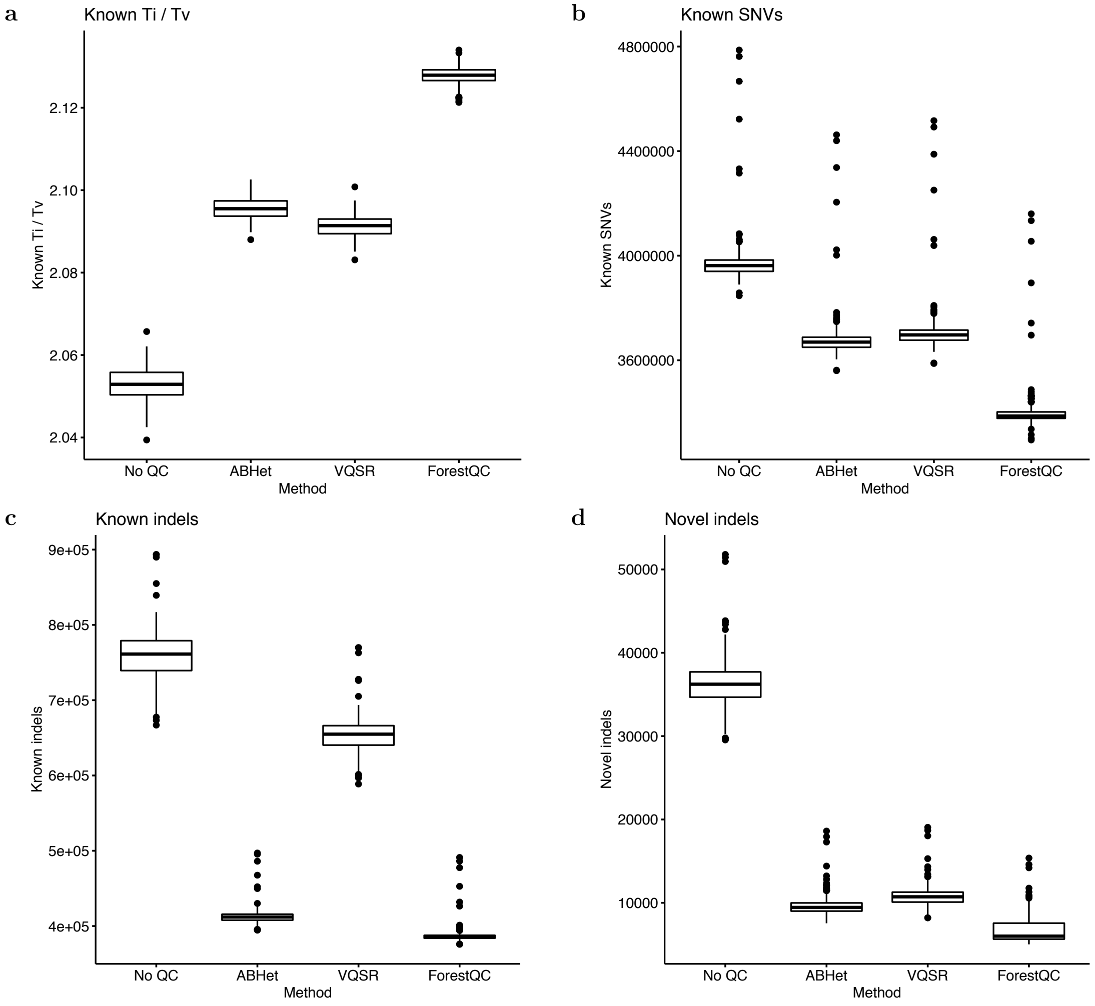
Supplementary Figure 7: Overall quality of rare variants (MAF < 0.03) and common variants (MAF ≥ 0.03) in the BP dataset. The average Mendelian error rate and genotype missing rate for SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.



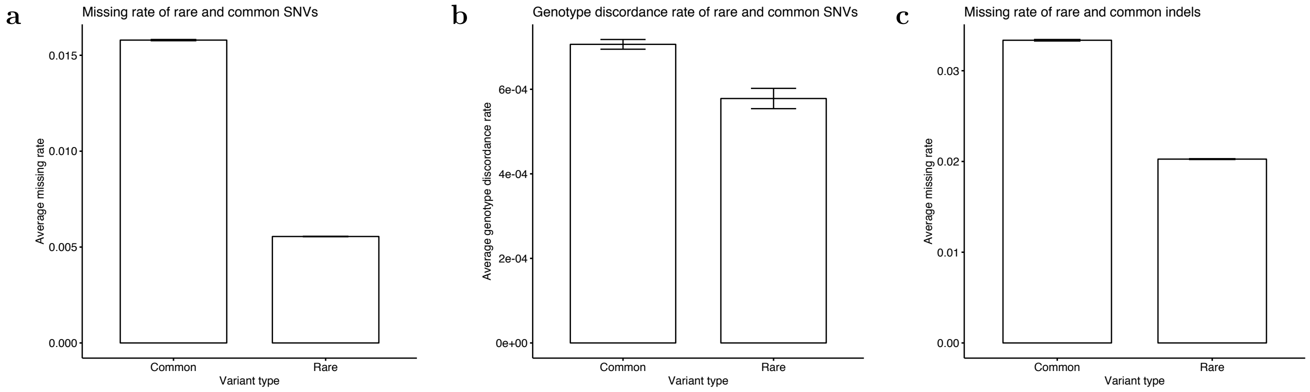
Supplementary Figure 8: (a) Ti/Tv ratio of SNVs not found in dbSNP, (b) the number of total SNVs and (c) the number of total indels in the BP dataset processed with VQSR using "SOR" or not. SOR stands for StrandOddsRatio, which is a metric for strand bias measured by the Symmetric Odds Ratio test. The version of dbSNP is 150.



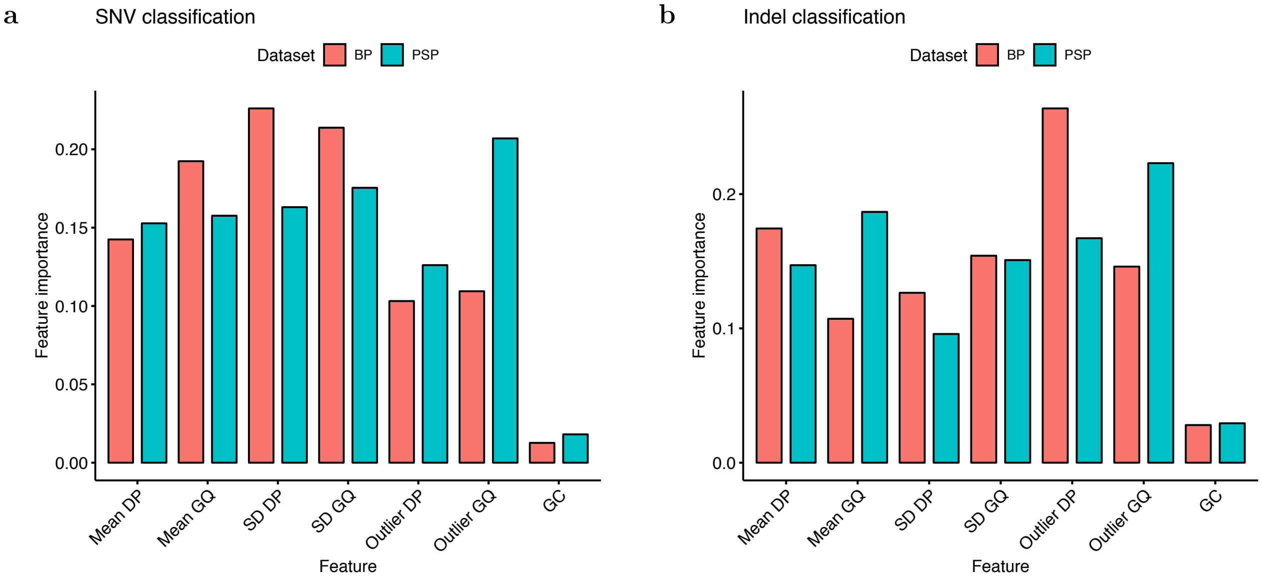
Supplementary Figure 9: Overall quality of bad variants in the PSP dataset detected by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. The average genotype missing rate for both SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.



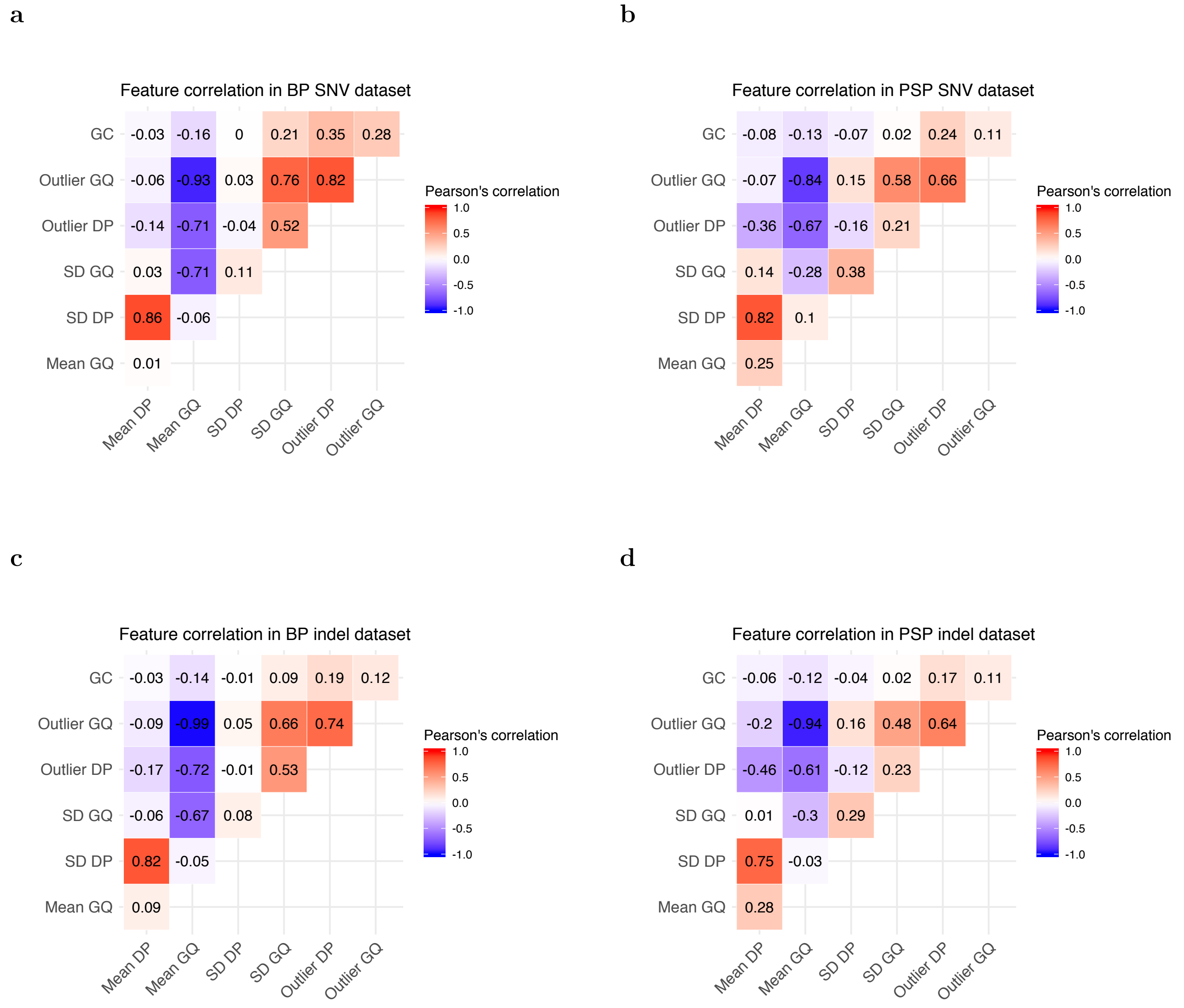
Supplementary Figure 10: Sample-level quality metrics of good variants in PSP dataset identified by four different methods, including no QC applied, ABHet approach, VQSR and ForestQC. (a) Ti/Tv ratio of SNVs found in dbSNP. (b) The number of SNVs found in dbSNP. (c) The number of indels found in dbSNP. (d) The number of indels not found in dbSNP. The version of dbSNP is 150.



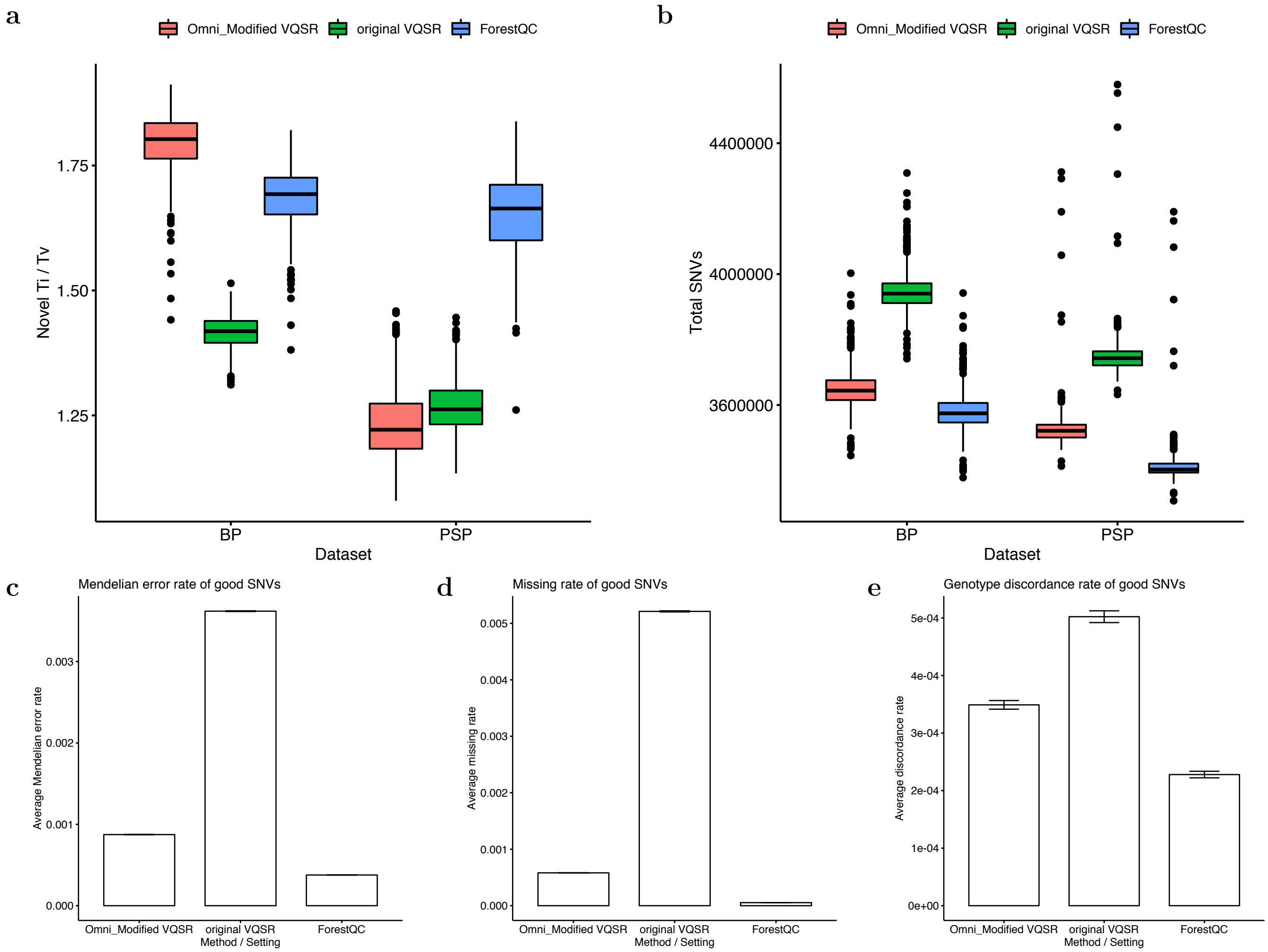
Supplementary Figure 11: Overall quality of rare variants (MAF < 0.03) and common variants (MAF $\geq$ 0.03) in the PSP dataset. The average genotype missing rate for SNVs and indels, and genotype discordance rate to microarray data for SNVs are shown.



Supplementary Figure 12: Feature importance of each feature in the random forest model of ForestQC applied to the BP and PSP datasets. DP stands for sequencing depth. GQ stands for genotyping quality. SD means standard deviation. Outlier DP or GQ means the proportion of samples having genotyping quality or sequencing depth lower than the first quartile of depth or genotyping quality in chromosome 1. GC stands for the GC content of a 1000-bp window where the variant is located. (a) Feature importance in SNV classification. (b) Feature importance in indel classification.



Supplementary Figure 13: Pearson's correlation coefficients between each pair of features in the BP and PSP dataset.



Supplementary Figure 14: Quality of good SNVs identified by VQSR with two different settings of training resources and ForestQC. (a) Ti/Tv ratio of SNVs not found in dbSNP v150 and (b) total number of SNVs in the BP and PSP dataset. (c)-(e) Average Mendelian error rate, average genotype missing rate, and average genotype discordance rate of good SNVs in the BP dataset. "Omni_Modified VQSR": SNVs in Omni chip array call set are considered to contain both true and false positive sites. "original VQSR": SNVs in Omni chip array call set are considered to contain only true sites.

Filter	Threshold
Mendelian error rate	0
Missing rate	$< 0.5\%$
HWE p-value	> 0.01
ABHet	0.3 - 0.7

Supplementary Table 1: Thresholds of four filters for the selection of good variants from the original dataset. Each good variant must satisfy all thresholds.

Condition	Filter	Rare variants (MAF < 0.03)	Common variants (MAF ≥ 0.03)
ALL	Mendelian error rate	> 3 / (# of trios)	> 5 / (# of trios)
	Missing rate	> 2%	> 3%
	HWE p-value	< 0.005	< 0.0005
	ABHet	> 0.75 or < 0.25	> 0.75 or < 0.25
ANY	Mendelian error rate	> 8 / (# of trios)	> 10 / (# of trios)
	Missing rate	> 8%	> 10%
	HWE p-value	< 0.001	< 1e-8

Supplementary Table 2: Thresholds of four filters for the selection of bad variants from the original dataset. ALL means all thresholds should be satisfied. ANY means variants are considered bad if they satisfy any one of the thresholds. Note that rare variants (MAF < 0.03) and common variants (MAF ≥ 0.03) have different thresholds.

Metric	ForestQC	ForestQC (No ME)
Total SNVs	22,227,503	22,301,653
Known SNVs	19,361,635	19,401,450
Known SNVs (%)	87.11%	87.00%
Novel SNVs	2,865,868	2,900,203
Novel SNVs (%)	12.89%	13.00%
Known Ti / Tv	2.1678	2.1620
Novel Ti / Tv	1.7790	1.7577
Total indels	2,789,037	2,813,369
Known indels	2,237,002	2,251,421
Known indels (%)	80.21%	80.03%
Novel indels	552,035	561,948
Novel indels (%)	19.79%	19.97%
Multi-allelic SNVs	77,693	78,220
Multi-allelic SNVs (%)	0.35%	0.35%
Known multi-allelic SNVs	75,107	75,378
Known multi-allelic SNVs (%)	0.39%	0.39%
Singletons in SNVs	3,801,389	3,804,176
Singletons in SNVs (%)	17.10%	17.06%
Singletons in indels	433,222	433,035
Singletons in indels (%)	15.53%	15.39%

Supplementary Table 3: Sample-level quality metrics of variants in the BP dataset processed by ForestQC using ME rate as a filter or not. There are 20 metrics in total, which are described in Methods section in detail. “Known” stands for variants in dbSNP. “Novel” stands for variants not in dbSNP. The version of dbSNP is 150.

Method	Rare SNVs	Common SNVs	Rare indels	Common indels
No QC	16,304,019 (65.00%)	8,777,577 (35.00%)	2,219,301 (55.81%)	1,757,402 (44.19%)
ABHet	14,682,507 (65.50%)	7,732,821 (34.50%)	1,655,360 (61.98%)	1,015,280 (38.02%)
VQSR	15,841,775 (65.65%)	8,288,407 (34.35%)	1,700,244 (54.12%)	1,441,564 (45.88%)
ForestQC	14,952,779 (67.27%)	7,274,684 (32.73%)	1,876,432 (67.28%)	912,598(32.72%)

Supplementary Table 4: The number and fraction of rare variants (MAF < 0.03) and common variants (MAF ≥ 0.03) in all good variants identified by different methods in the BP dataset.

Method	Rare SNVs	Common SNVs	Rare indels	Common indels
No QC	24,864,011 (74.73%)	8,409,100 (25.27%)	3,381,339 (66.39%)	1,712,104 (33.61%)
ABHet	22,005,560 (75.04%)	7,321,250 (24.96%)	2,337,310 (72.66%)	879,481 (27.34%)
VQSR	23,593,775 (75.42%)	7,687,845 (24.58%)	2,349,383 (63.80%)	1,332,936 (36.20%)
ForestQC	22,525,090 (76.74%)	6,827,239 (23.26%)	2,603,084 (76.15%)	815,158 (23.85%)

Supplementary Table 5: The number and fraction of rare variants (MAF < 0.03) and common variants (MAF $\geq$ 0.03) in all good variants identified by different methods in PSP dataset.

Method	BP SNV	BP indel	PSP SNV	PSP indel
ForestQC	17.00 min	3.74 min	23.24 min	5.82 min
GATK-VQSR	6.03 h	1.21 h	8.30 h	1.44 h

Supplementary Table 6: Running time of ForestQC and VQSR in the BP and PSP datasets, measured in real time.

Metric	Definition	Sample-level or Variant-level
Het / Hom	(count of heterozygous calls) / (count of homozygous non-reference calls)	Sample-level only
Total SNVs	Number of SNVs calls (i.e. non-reference genotypes) that were examined	Both
Total SNVs (%)	The proportion of total SNVs of a sample in total SNVs of the entire variant dataset	Sample-level only
Known SNVs	Number of SNVs found in dbSNP	Both
Known SNVs (%)	The proportion of SNVs in dbSNP	Both
Novel SNVs	Number of SNVs not in dbSNP	Both
Novel SNVs (%)	The proportion of SNVs not in dbSNP	Both
Known Ti / Tv	The Ti / Tv ratio of the SNV calls made at dbSNP sites	Both
Novel Ti / Tv	The Ti / Tv ratio of the SNV calls made at non-dbSNP sites	Both
Total indels	Number of indels that were examined	Both
Total indels (%)	The proportion of total indels of a sample in total indels of the entire variant dataset	Sample-level only
Known indels	Number of indels in dbSNP	Both
Known indels (%)	The proportion of indels in dbSNP	Both
Novel indels	Number of indels not in dbSNP	Both
Novel indels (%)	The proportion of indels not in dbSNP	Both
Multi-allelic SNVs	Number of multi-allelic SNVs	Variant-level only
Multi-allelic SNVs (%)	The proportion of multi-allelic SNVs	Variant-level only
Known multi-allelic SNVs	Number of multi-allelic SNVs in dbSNP	Both
Known multi-allelic SNVs (%)	The proportion of multi-allelic SNVs in dbSNP	Both
Singletons in SNVs	Number of singletons in SNVs	Both
Singletons in SNVs (%)	The proportion of singletons in SNVs	Both
Singletons in indels	Number of singletons in indels	Both
Singletons in indels (%)	The proportion of singletons in indels	Both

Supplementary Table 7: Definition of 23 metrics for sequencing quality control calculated for sample-level and variant-level. Only three metrics, (Het / Hom, % Total SNVs and % total indels) are only calculated for sample-level. Other metrics are measured for every variant site and every sample. The version of dbSNP used in this study is 150.